

BENJAMINS

■

TRANSLATION

Computers and
Translation

edited by
Harold Somers

N · LIBRARY

Computers and Translation

Benjamins Translation Library

The Benjamins Translation Library aims to stimulate research and training in translation and interpreting studies. The Library provides a forum for a variety of approaches (which may sometimes be conflicting) in a socio-cultural, historical, theoretical, applied and pedagogical context. The Library includes scholarly works, reference works, post-graduate text books and readers in the English language.

General editor

Gideon Toury
Tel Aviv University

Associate editor

Miriam Shlesinger
Bar Ilan University

Advisory board

Marilyn Gaddis Rose
Binghamton University

Yves Gambier
Turku University

Daniel Gile
Université Lumière Lyon 2 and ISIT Paris

Ulrich Heid
University of Stuttgart

Eva Hung
Chinese University of Hong Kong

W. John Hutchins
University of East Anglia

Zuzana Jettmarová
Charles University of Prague

Werner Koller
Bergen University

Alet Kruger
UNISA

José Lambert
Catholic University of Leuven

Franz Pöchhacker
University of Vienna

Rosa Rabadán
University of León

Roda Roberts
University of Ottawa

Juan C. Sager
UMIST Manchester

Mary Snell-Hornby
University of Vienna

Sonja Tirkkonen-Condit
University of Joensuu

Lawrence Venuti
Temple University

Wolfram Wilss
University of Saarbrücken

Judith Woodsworth
Mt. Saint Vincent University Halifax

Sue Ellen Wright
Kent State University

Volume 35

Computers and Translation: A translator's guide
Edited by Harold Somers

Computers and Translation

A translator's guide

Edited by

Harold Somers

UMIST

John Benjamins Publishing Company
Amsterdam/Philadelphia



TM The paper used in this publication meets the minimum requirements of American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

Library of Congress Cataloging-in-Publication Data

Computers and translation : a translator's guide / edited by Harold Somers.

p. cm. (Benjamins Translations Library, ISSN 0929-7316 ; v. 35)

Includes bibliographical references and indexes.

1. Machine translating. I. Somers, H.L. II. Series.

P308. C667 2003

418'.02'0285-dc21

2003048079

ISBN 90 272 1640 1 (Eur.) / 1 58811 377 9 (US) (Hb; alk. paper)

© 2003 – John Benjamins B.V.

No part of this book may be reproduced in any form, by print, photoprint, microfilm, or any other means, without written permission from the publisher.

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands
John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

*For Mum, Happy 81st
and Dad, the first linguist I ever met,
who (I hope) would have found all this fascinating
and for Nathan and Joe, the next generation*

Table of contents

List of figures	ix
List of tables	xiii
List of contributors	xv
CHAPTER 1	
Introduction	1
<i>Harold Somers</i>	
CHAPTER 2	
The translator's workstation	13
<i>Harold Somers</i>	
CHAPTER 3	
Translation memory systems	31
<i>Harold Somers</i>	
CHAPTER 4	
Terminology tools for translators	49
<i>Lynne Bowker</i>	
CHAPTER 5	
Localisation and translation	67
<i>Bert Esselink</i>	
CHAPTER 6	
Translation technologies and minority languages	87
<i>Harold Somers</i>	
CHAPTER 7	
Corpora and the translator	105
<i>Sara Laviosa</i>	
CHAPTER 8	
Why translation is difficult for computers	119
<i>Doug Arnold</i>	

CHAPTER 9	
The relevance of linguistics for machine translation	143
<i>Paul Bennett</i>	
CHAPTER 10	
Commercial systems: The state of the art	161
<i>John Hutchins</i>	
CHAPTER 11	
Inside commercial machine translation	175
<i>Scott Bennett and Laurie Gerber</i>	
CHAPTER 12	
Going live on the internet	191
<i>Jin Yang and Elke Lange</i>	
CHAPTER 13	
How to evaluate machine translation	211
<i>John S. White</i>	
CHAPTER 14	
Controlled language for authoring and translation	245
<i>Eric Nyberg, Teruko Mitamura and Willem-Olaf Huijsen</i>	
CHAPTER 15	
Sublanguage	283
<i>Harold Somers</i>	
CHAPTER 16	
Post-editing	297
<i>Jeffrey Allen</i>	
CHAPTER 17	
Machine translation in the classroom	319
<i>Harold Somers</i>	
Index	341

List of figures

Chapter 2

1.	<i>Transit</i> : An example of a translator's workstation	15
2.	Translating in-figure captions can be easier	18
3.	Online version of Langenscheidt's <i>New College Dictionary</i> (from the <i>T1 Professional</i> system)	20
4.	Dictionary entry shown by clicking on link in Figure 3	21
5.	Adding to a dictionary entry (from the <i>French Assistant</i> system)	22
6.	Word-processor with additional menus and toolbars (from the <i>Trados</i> system)	22
7.	Source and target text in parallel windows (from <i>French Assistant</i>)	23
8.	Interactive translation (<i>French Assistant</i>)	24
9.	Concordance of the word <i>curious</i> in <i>Alice's Adventures in Wonderland</i>	25
10.	An English–Japanese bilingual concordance listing for the word <i>Translator's (Trados)</i>	26
11.	Bilingual concordance of the phrase <i>point of order</i> in the Canadian Hansard	27
12.	Bilingual concordance of the word-pair <i>librairie–library</i> in the Canadian Hansard	27
13.	Bilingual concordance of the word <i>rise</i> in the Canadian Hansard	28

Chapter 3

1.	<i>Trados</i> 's translation memory window showing partial match	31
2.	A similar feature in Atril's <i>Déjà Vu</i> system	32
3.	Output of an alignment tool	36
4.	IBM's <i>Translation Manager</i> showing multiple matches	38
5.	"Portion matching" in <i>Déjà Vu</i>	41

Chapter 4

1.	Conventional TMSs came with a fixed set of pre-defined fields	54
2.	Flexible TMSs, such as <i>TermBase</i> from MultiCorpora, allow translators to create and organize their own information fields	54
3.	Term records retrieved using fuzzy matching	55

4. Sample hit lists retrieved for different search patterns	56
5. Automatic terminology lookup in <i>Trados</i>	56
6. A hybrid text produced as a result of pre-translation in <i>Trados</i>	57
7. Multiple forms of the term can be recorded on a term record to facilitate automatic insertion of the required form directly into the target text	59
Chapter 5	
1. A dialog box localised for Swedish	71
2. Drop-down menu showing hot keys	72
3. The <i>Passolo</i> software localisation system	82
Chapter 6	
1. English QWERTY (above) and French AZERTY (below) keyboard layouts	91
2. Arabic keyboard	92
3. Justification in Arabic achieved by stretching the letter forms	93
Chapter 7	
1. Types of translation corpus	106
Chapter 8	
1. The “pyramid” diagram	123
Chapter 11	
1. Typically, the greater the degree of automation in system development (learning of analysis and translation rules), the shallower the analysis the system performs. In the extreme case, learning is fully automated, and the system uses no conventional grammar or lexicon	179
Chapter 12	
1. <i>Babelfish</i> front page as it appeared in November 2002	192
2. Search results including “Translate” button	192
3. Translation button included in web page	193
4. Technical configuration of <i>babelfish</i> service (Story, 1998)	194
5. Feedback panel in <i>babelfish</i> web-page	196
6. Distribution of language pairs	204
7. Screen capture of multilingual chat hosted by Amikai.com	207
8. The same chat as seen from another perspective	208

Chapter 13

1.	Case 1: counting errors	215
2.	Case 2: intelligibility and fidelity	217
3.	Case 3: before and after	218
4.	Internal representation of (wrong) syntactic analysis of (7a)	226
5.	Example of radar chart resulting from questionnaire	234
6.	Example of JEIDA radar chart corresponding to a given system type	234
7.	Example of an adequacy evaluation page, from a 1994 evaluation	237
8.	Example of fluency evaluation page, from a recent evaluation	237

Chapter 14

1.	Examples of Simplified English: <i>prevent</i> vs. <i>preventive</i> and <i>right</i> vs. <i>right-hand</i>	246
2.	CL Checking and Translation in <i>KANT</i>	260

Chapter 15

1.	Examples of movement words in stock-market reports (from Kittredge, 1982:118)	285
2.	Weather report as received	290

Chapter 16

1.	Changes to ECTS texts learned by the APE module	314
----	---	-----

Chapter 17

1.	Semantic attributes for new dictionary entry	324
2.	<i>TransIt-TIGER</i> in “Hints” mode	330
3.	Example of Russian web page	331
4.	<i>Babelfish</i> ’s translation of text in Figure 3	332

List of tables

Chapter 6

- | | | |
|----|---|----|
| 1. | Provision of computational resources for some “exotic” languages
of relevance to the situation in the UK | 90 |
|----|---|----|

Chapter 11

- | | | |
|----|---|-----|
| 1. | Abstract data structures for sentence (1) | 177 |
|----|---|-----|

Chapter 12

- | | | |
|----|---|-----|
| 1. | Total number of translations on two census days | 203 |
| 2. | Translation type (Text vs. Web-page) | 203 |
| 3. | Length of texts submitted for translation | 204 |

List of contributors

Jeffrey Allen, Mycom France, Paris, France.

Postediting@aol.com

Doug Arnold, Department of Language and Linguistics, University of Essex,
Wivenhoe Park, Colchester CO4 3SQ, England. doug@essex.ac.uk

Paul Bennett, Centre for Computational Linguistics, UMIST, PO Box 88,
Manchester M60 1QD, England. paul.bennett@umist.ac.uk

Scott Bennett, 43 West Shore Road, Denville, NJ 07834, USA.
three.bennetts@verizon.net

Lynne Bowker, School of Translation and Interpretation, University of Ottawa, 70
Laurier Ave E., PO Box 450, Station A, Ottawa ON K1N 6N5, Canada.
lbowker@uottawa.ca

Bert Esselink, L10nbridge, Overschiestraat 55, 1062 HN Amsterdam, The Netherlands.
bert@locguide.com

Laurie Gerber, Language Technology Broker, 4774 Del Mar Avenue, San Diego
CA, USA. lgerber@gerbersite.com

Willem-Olaf Huijsen, Institute for Linguistics OTS, Utrecht University, Trans 10,
3512 JK, Utrecht, The Netherlands. willem-olaf.huijsen@let.ruu.nl

John Hutchins, University of East Anglia, Norwich NR4 7TJ, England.
wjhutchins@compuserve.com

Elke Lange, SYSTRAN Software, Inc., 9333 Genesee Avenue, San Diego CA 92121,
USA. elange@systransoft.com

Sara Laviosa, Università degli Studi di Bari, Italy. SaraLaviosa@hotmail.com

Teruko Mitamura, Language Technologies Institute, Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh PA 15213, USA. teruko+@cs.cmu.edu

Eric Nyberg, Language Technologies Institute, Carnegie Mellon University, 5000
Forbes Ave, Pittsburgh PA 15213, USA. ehn@cs.cmu.edu

Harold Somers, Centre for Computational Linguistics, UMIST, PO Box 88,
Manchester M60 1QD, England. harold.somers@umist.ac.uk

John S. White, PRC Northrop Grumman Information Technology, MacLean VA,
USA. white_john@prc.com

Jin Yang, SYSTRAN Software, Inc., 9333 Genesee Avenue, San Diego CA 92121,
USA. jyang@systransoft.com

CHAPTER 1

Introduction*

Harold Somers
UMIST, Manchester, England

1. Preliminary remarks

This book is, broadly speaking, and as the title suggests, about computers and translators. It is not, however, a Computer Science book, nor does it have *much* to say about Translation Theory. Rather it is a book for translators and other professional linguists (technical writers, bilingual secretaries, language teachers even), which aims at clarifying, explaining and exemplifying the impact that computers have had and are having on their profession. It is about Machine Translation (MT), but it is also about Computer-Aided (or -Assisted) Translation (CAT), computer-based resources for translators, the past, present and future of translation and the computer.

Actually, there is a healthy discussion in the field just now about the appropriateness or otherwise of terms like the ones just used. The most widespread term, “Machine Translation”, is felt by many to be misleading (who calls a computer a “machine” these days?) and unhelpful. But no really good alternative has presented itself. Terms like “translation technology” or “translation software” are perhaps more helpful in indicating that we are talking about computers, the latter term emphasising that we are more interested in computer programs than computer hardware as such. Replacing the word “translation” by something like “translator’s” helps to take the focus away from translation as the end product and towards translation as a process¹ carried out by a human (the translator) using various tools, among which we are interested in only those that have something to do with computers.

We hope that this book will show you how the computer can help you, and in doing so we hope to show also what the computer *cannot* do, and thereby reassure you that the computer, far from being a threat to your livelihood, can become an essential tool which will make your job easier and more satisfying.

1.1 Who are we?

This book has been put together by academics (teachers and researchers in language and linguistics, especially computational linguistics, translation theory), employees of software companies, and — yes — even translators. All the contributors have an interest in the various aspects of translation and computers, and between them have several hundred years' worth of experience in the field. All are committed to telling a true story about computers and translation, what they can and cannot do, what they are good for, and what they are not. We are *not* trying to sell you some product. But what we *are* aiming to do is to dispel some of the myths and prejudices that we see and hear on translators' forums on the Internet, in the popular press, even in books about translation whose authors should know better!

1.2 Who are you?

We assume that you are someone who knows about and is interested in languages and translation. Perhaps you are a professional linguist, or would like to be. Or perhaps you are just a keen observer. In particular, you are interested in the topic of computers and translation and not too hostile, though perhaps healthily sceptical. The fact you have got hold of this book (perhaps you have already bought it, or are browsing in a bookshop, or a colleague has passed it on to you) is taken to mean that you have not dismissed the idea that computers can play a part in the translation process, and are open to some new ideas.

You are probably *not* a computer buff: if you are looking for lots of stuff about bits and bytes, integer float memory and peripheral devices then this is not the book for you. On the other hand, you are probably a regular computer-*user*, perhaps at the level of word-processing and surfing the World Wide Web. You know, roughly, the difference between “software” and “hardware”, you know about windows and desktops, files and folders. You may occasionally use the computer to play games, and you may even have used some software that involves a kind of programming or authoring. But by enlarge that's not really your area of expertise.

On the other hand, you *do* know about language. We don't need to tell you about how different languages say things differently, about how words don't always neatly correspond in meaning and use, and how there's almost never an easy answer to the question “How do you say X in language Y?” (though we may remind you from time to time). We assume that you are familiar with traditional grammatical terminology (noun, verb, gender, tense, etc.) though you may not have studied linguistics as such. Above all, we don't need to remind you that translation is an art, not a science, that there's no such thing as a single “correct” translation, that a

translator’s work is often under-valued, that translation is a human skill — one of the oldest known to humankind² — not a mechanical one. Something else you already know is that almost no one earns their living translating literary works and poetry: translation is mostly technical, often nonetheless demanding, but just as often routine and sometimes — dare we admit it? — banal and boring. Whatever the case, the computer has a role to play in your work.

1.3 Conventions in this book

This is a technical book, and as such will, we hope, open avenues of interest for the reader. For that reason, we give references to the literature to support our arguments, in the usual academic fashion. Where specific points are made, we use footnotes so as to avoid cluttering the text with unwieldy references. We also want to direct the reader to further sources of information, which are gathered together at the end of each chapter. Technical terms are introduced in bold font. Software product names are given in italics, and are thus distinguished typographically from the (often identical) names of the company which produce them.

Often it is necessary to give language examples to illustrate the point being made. We follow the convention of linguistics books as follows: **cited forms** are always given in italics, regardless of language. Meanings or **glosses** are given in single quotes. Cited forms in languages other than English are always accompanied by a **literal gloss** and/or a translation, as appropriate, unless the meaning is obvious from the text. Thus, we might write that *key-ring* is rendered in Portuguese as *porta-chave* lit. ‘carry-key’, or that in German the plural of *Hund* ‘dog’ is *Hunde*. Longer examples (phrases and sentences) are usually separated from the text and referred to by a number in brackets, as in (1). Foreign-language examples are accompanied by an aligned literal gloss as well as a translation (2a), though either may be omitted if the English follows the structure of the original closely enough (2b).

- (1) This is an example of an English sentence.
- (2)
 - a. *Ein Lehrbuchbeispiel in deutscher Sprache ist auch zu geben.*
a text-book-example in German language is also to give
'A German-language example from a text-book can also be given.'
 - b. *Voici une phrase en français.*
this-is a sentence in French

We follow the usual convention from linguistics of indicating with an asterisk that a sentence or phrase is **ungrammatical** or otherwise **anomalous** (3a), and a question-mark if the sentence is dubious (3b).

- (3) a. *This sentence are wrong.
b. ?Up with this we will not put.

2. Historical sketch

A mechanical translation tool has been the stuff of dreams for many years. Often found in modern science fiction (the universal decoder in *Star Trek*, for example), the idea predates the invention of computers by a few centuries. Translation has been a suggested use of computers ever since they were invented (and even before, curiously). Universal languages in the form of numerical codes were proposed by several philosophers in the 17th Century, most notably Leibniz, Descartes and John Wilkins.

In 1933 two patents had been independently issued for “translation machines”, one to Georges Artsrouni in France, and the other to Petr Petrovich Smirnov-Troyanskii in the Soviet Union. However, the history of MT is usually said to date from a period just after the Second World War during which computers had been used for code-breaking. The idea that translation might be in some sense similar at least from the point of view of computation is attributed to Warren Weaver, at that time vice-president of the Rockefeller Foundation. Between 1947 and 1949, Weaver made contact with a number of colleagues in the USA and abroad, trying to raise interest in the question of using the new digital computers (or “electronic brains” as they were popularly known) for translation; Weaver particularly made a link between translation and cryptography, though from the early days most researchers recognised that it was a more difficult problem.

2.1 Early research

There was a mixed reaction to Weaver’s ideas, and significantly MIT decided to appoint Yehoshua Bar-Hillel to a full-time research post in 1951. A year later MIT hosted a conference on MT, attended by 18 individuals interested in the subject. Over the next ten to fifteen years, MT research groups started work in a number of countries: notably in the USA, where increasingly large grants from government, military and private sources were awarded, but also in the USSR, Great Britain, Canada, and elsewhere. In the USA alone at least \$12 million and perhaps as much as \$20 million was invested in MT research.

In 1964, the US government decided to see if its money had been well spent, and set up the Automated Language Processing Advisory Committee (ALPAC). Their report, published in 1966, was highly negative about MT with very damaging consequences. Focussing on Russian–English MT in the USA, it concluded that MT was slower, less accurate and twice as expensive as human translation, for which

there was in any case not a huge demand. It concluded, infamously, that there was “no immediate or predictable prospect of useful machine translation”. In fact, the ALPAC report went on to propose instead fundamental research in computational linguistics, and suggested that *machine-aided* translation may be feasible. The damage was done however, and MT research declined quickly, not only in the USA but elsewhere.

Actually, the conclusions of the ALPAC report should not have been a great surprise. The early efforts at getting computers to translate were hampered by primitive technology, and a basic under-estimation of the difficulty of the problem on the part of the researchers, who were mostly mathematicians and electrical engineers, rather than linguists. Indeed, theoretical (formal) linguistics was in its infancy at this time: Chomsky’s revolutionary ideas were only just gaining widespread acceptance. That MT was difficult was recognised by the likes of Bar-Hillel who wrote about the “semantic barrier” to translation several years before the ALPAC committee began its deliberations, and proposals for a more sophisticated approach to MT can be found in publications dating from the mid- to late-1950s.

2.2 “Blind idiots”, and other myths

It is at about this time too that much repeated (though almost certainly apocryphal) stories about bad computer-generated translations became widespread. Reports of systems translating *out of sight, out of mind* into the Russian equivalent of *blind idiot*, or *The spirit is willing but the flesh is weak* into *The vodka is good but the meat is rotten* can be found in articles about MT in the late 1950s; looking at the systems that were around at this period one has difficulty in imagining any of them able to make this kind of quite sophisticated mistranslation, and some commentators (the present author included) have suggested that similar stories have been told about incompetent *human* translators.

2.3 The “second generation” of MT systems

The 1970s and early 1980s saw MT research taking place largely outside the USA and USSR: in Canada, western Europe and Japan, political and cultural needs were quite different. Canada’s bilingual policy led to the establishment of a significant research group at the University of Montreal. In Europe groups in France, Germany and Italy worked on MT, and the decision of the Commission of the European Communities in Luxembourg to experiment with the *Systran* system (an American system which had survived the ALPAC purge thanks to private funding) was highly significant. In Japan, some success with getting computers to handle the complex writing system of Japanese had encouraged university and industrial

research groups to investigate Japanese–English translation.

Systems developed during this period largely share a common design basis, incorporating ideas from structural linguistics and computer science. As will be described in later chapters, system design divided the translation problem into manageable sub-problems — analysing the input text into a linguistic representation, adapting the source-language representation to the target language, then generating the target-language text. The software for each of these steps would be separated and modularised, and would consist of grammars developed by linguists using formalisms from theoretical linguists rather than low-level computer programs. The lexical data (*ictionaries*) likewise were coded separately in a transparent manner, so that ordinary linguists and translators could work on the projects, not needing to know too much about how the computer programs actually worked.

2.4 Practical MT systems

By the mid 1980s, it was generally recognised that fully automatic high-quality translation of unrestricted texts (FAHQT) was not a goal that was going to be readily achievable in the near future. Researchers in MT started to look at ways in which usable and useful MT systems could be developed even if they fell short of this goal. Many commentators now distinguish between the use of MT for **assimilation**, where the user is a reader of a text written in an unfamiliar language, and **dissemination**, where the user is the author of a text to be published in one or more languages. In particular, the idea that MT could work if the input text was somehow *restricted* gained currency. This view developed as the **sublanguage** approach, where MT systems would be developed with some specific application in mind, in which the language used would be a subset of the “full” language, hence “sublanguage”³ (see Chapter 15). This approach is especially seen in the highly successful *Météo* system, developed at Montreal, which was able to translate weather bulletins from English into French, a task which human translators obviously found very tedious. Closely related to the sublanguage approach is the idea of using **controlled language**, as seen in technical authoring (see Chapter 14).

The other major development, also in response to the difficulty of FAHQT, was the concept of computer-based *tools* for translators, in the form of the **translator’s workstation** (see Chapter 2). This idea was further supported by the emergence of small-scale inexpensive computer hardware (“microcomputers”, later more usually known as personal computers, PCs). Here, the translator would be provided with software and other computer-based facilities to assist in the task of translation, which remained under the control of the human: Computer-Aided (or -Assisted) Translation, or CAT. These tools would range in sophistication, from the (nowadays almost ubiquitous) multilingual word-processing, with spell checkers, synonym

lists (“thesauri”) and so on, via on-line dictionaries (mono- and multilingual) and other reference sources, to machine-aided translation systems which might perform a partial draft translation for the translator to tidy up or **post-edit**. As computers have become more sophisticated, other tools have been developed, most notably the **translation memory** tool which will be familiar to many readers (see Chapter 3).

2.5 Latest research

Coming into the 1990s and the present day, we see MT and CAT products being marketed and used (and, regrettably sometimes misused) both by language professionals and by amateurs, the latter for translating e-mails and World Wide Web pages. This use will of course be the subject of much of the rest of this book. Meanwhile, MT researchers continue to set themselves ambitious goals.

Spoken-language translation (SLT) is one of these goals. SLT combines two extremely difficult computational tasks: speech understanding, and translation. The first task involves extracting from an acoustic signal the relevant bits of sound that can be interpreted as speech (that is, ignoring background noise as well as vocalisations that are not speech as such), correctly identifying the individual speech sounds (phonemes) and the words that they comprise and then filtering out distractions such as hesitations, repetitions, false starts, incomplete sentences and so on, to give a coherent text message. All this then has to be translated, a task quite different from that of translating written text, since often it is the content rather than the form of the message that is paramount. Furthermore, the constraints of real-time processing are a considerable additional burden. Try this experiment next time you are in a conversation: count to 5 under your breath before replying to any remark, even the most simple or banal, *Good morning*, or whatever. Your conversation partner will soon suspect something is wrong, especially if you try this over the telephone! But given the current state of the art in SLT, a system that could process the input and give a reasonable translation — in synthesised speech of course — within 5 seconds would be considered rather good.

Turning back to MT for written text, another concern is coverage of a wider variety of languages. So far, it is mainly the commercially important languages of western Europe and the Far East that have received the attention of developers. It is recognised that there are thousands more languages in the world for which MT or at least CAT software should be developed, but each new language pair poses huge problems, even if work done on a similar or related language can be used as a starting point. If a classical (linguistic) approach to MT is taken, then grammar rules for the new language must be written, “transfer” rules between the new and old languages and, the biggest bottle-neck of all, dictionary entries for thousands of words must be written. Even though the dictionaries used by MT systems contain information that

is rather different — and in a different format — from conventional dictionaries, the latter can sometimes be used, if they are in a **machine-readable form** (i.e. can be input into the computer), to extract the relevant information. Another technique that is starting to be explored is the extraction of linguistic information from large **parallel corpora**, that is, collections of texts together with their translations, assuming that the two “sides” of the corpus can be neatly “aligned”.

These research issues may not, however, be of immediate relevance to the reader of this book at the moment (though watch out in the near future!). In compiling this collection, we have preferred to focus on the current state of practical and usable MT and CAT, and in the final section of this introductory chapter, we provide a brief overview of the contents of this book.

3. Overview of this book

The first seven chapters look at various uses to which a translator might put the computer while the second half of the book focuses more on MT. In Chapter 2 we describe the development of the ideas behind the **translator’s workstation**, and look at some of the computer-based tools that can be made easily available to translators, with a special focus in Chapter 3 on one of these tools, the **translation memory**. Chapter 4 concerns the special place of **terminology** in the CAT scenario. Translators have always been aware of the need to access technical vocabulary and be sure that the terms chosen are correct and appropriate. As Lynne Bowker describes, computers can play a particularly useful role in this question, as **term banks** and other sources of terminology are available in various forms, both on-line and in the form of machine-readable dictionaries and thesauri.

A relatively new translation activity that has emerged in recent years goes under the name of **software localization**. In the early days of computers, most software (and hardware) that was produced was biased towards (American) English-speaking users. It has now been recognised that products aimed at a global market must be customized for local aspects of that global market. Software localization involves translating documentation, including on-line help files, but also often involves customizing the software itself, inasmuch as it contains language (for example, how to translate *Press Y for Yes* into French). In Chapter 5, Bert Esselink condenses some of the ideas from his comprehensive book on the subject, to give an indication of the problems involved, and some of the tools available to assist the translator in this specific task.

In today’s commercially-oriented world, much translation work is motivated by commercial considerations. Socio-economic factors thus influence the development of MT and CAT systems, and it is the major European languages (English,

French, Spanish, German, Italian, Portuguese, Russian) plus Japanese, Chinese, Korean and to a certain extent Arabic that have received attention from the developers. Bad luck if you work into (or out of) any of the several thousand other languages of the world. In Chapter 6 we look at the case of CAT and **minority languages** — an ironic term when one considers that the list of under-resourced languages includes several of the world's top 20 most spoken languages (Hindi, Bengali, Malay/Indonesian, Urdu, Punjabi, Telegu, Tamil, Marathi, Cantonese). We will consider what the prospects are for translators working in these languages (and other languages more reasonably described as “minority”) and what kinds of computer-based tools and resources could be made available.

Our next chapter looks at the place of computers in the academic world of translator training. Sara Laviosa considers the use of the computer in Translation Studies: in particular, this chapter looks at how computer-based **corpora** — collections of translated text — can be used to study trends in translation practice.

The remaining chapters focus more closely on MT. In Chapter 8, Doug Arnold explains **why translation is hard for a computer**. Readers of this book will have views on what aspects of translation are hard for humans, but Arnold points out that some aspects of language understanding in the first place, and then the rendering of what has been understood in a foreign language in the second place, present difficulties to computers which, after all, are basically sophisticated adding machines. At least some of the difficulty is down to the nature of language itself, and in Chapter 9, Paul Bennett describes how the scientific study of language — **linguistics** — can help to provide some solutions to the problems.

The next three chapters focus on MT from the commercial point of view. In Chapter 10, John Hutchins, MT's unofficial historian and archivist-in-chief, details the current state of the art in **commercially available** MT and CAT software. Chapter 11 presents the developer's point of view. Co-authored by Laurie Gerber, formerly one of Systran's senior linguists, and Scott Bennett who, at the time of writing was a senior member of Logos's development team, and before that had helped oversee the productization of the *Metal* MT system by Siemens. In Chapter 12, Jin Yang and Elke Lange report on Systran's intriguing experiment in which they have made their MT system freely available on the **World Wide Web**. This contribution explains why the company is happy to see their product freely used, and reports on a period of close monitoring of the web-site, and users' feedback and opinions.

John White's chapter on how to **evaluate** MT will be essential reading for anyone thinking of MT or CAT tools as a solution to their translation needs, whether they be an individual freelancer, a small translation company or part of the translation department of a large company. White gives a practical and historical overview of what to evaluate, how to evaluate it, and, above all, some of the pitfalls to avoid.

The next three chapters address aspects of the practical use of MT. In Chapters 14 and 15 we look at two strategies for getting the best out of MT: Eric Nyberg, Terako Mitamura and Wolf Huijsen describe their approach to **controlled language**, explaining the basic idea behind the concept, and how it can be implemented within a translation scenario. An important feature of the controlled-language approach is the need to gain acceptance of the underlying idea from the authors of the texts to be translated, and to overcome the negative preconceptions of loss of author's creativity and the resulting sterility of textual form that the term controlled language inevitably invokes. While the controlled-language approach restricts the syntax and vocabulary of the texts to be translated in a pre- and prescriptive way, the **sublanguage** approach takes advantage of naturally occurring restrictions and preferences in the style and vocabulary of the texts. In Chapter 15 we look at the classical example of a successful sublanguage MT system — the Canadian *Météo* system — and consider whether this was a “one-hit wonder” or whether the success of this experiment points the way for future MT success stories.

In Chapter 16, Jeffrey Allen looks at the question of revising MT output, usually termed **post-editing** to distinguish it from the parallel task of revision often performed on human translations. Allen brings out some of the differences in these two tasks, and outlines some strategies and techniques to make the task easier and more efficient.

In the final chapter we consider the use of MT and CAT tools in the **teaching** of translation, both to trainee translators, and to language students in general.

Further reading

The history of MT up to the mid 1980s is thoroughly documented in Hutchins (1986). Some “classical” papers have been collected in Nirenburg et al. (2003). Good, though now somewhat out-of-date, introductions to MT are given by Hutchins and Somers (1992) and Arnold et al. (1994). For more recent developments, see Trujillo (1999), which is also recommended for readers wishing to go into more technical detail about MT. Kay et al. (1994) is a good introduction to the problems of spoken language translation. References for other more specific aspects are given in the relevant chapters.

For on-going information, there are a number of good sources. The journal *Machine Translation* (published by Kluwer) is probably the premier academic journal in the field. A more informal source of information is the International Association for Machine Translation (IAMT) and its monthly newsletter *MT News International*. Membership of the IAMT is via one of its three regional chapters,

AMTA in the Americas, EAMT in Europe, and AAMT for Asia and Australasia. All three maintain a web presence.⁴

Notes

* I am very grateful to Ingrid Meyer, Gideon Toury, Lynne Bowker, Nick Somers and Federico Gaspari who read and commented on earlier drafts of this book, and to Bertie Kaal and Isja Conen at Benjamins, for their support. Above all I am grateful to the contributors who have, er, contributed so nicely. It should be noted that contributions have been written in British or American (or indeed Canadian) English, according to individual authors' choices. All trademarks are hereby acknowledged.

1. It is an interesting quirk of the English language that the vocabulary in general fails to distinguish between a process and its result: cf. *painting*, *building*, *decoration*, *cooking*, *teaching*, etc.
2. Some years ago, my professor — a pioneer in the field of computers and translation — was planning a prestigious talk he had been invited to give, and wanted my opinion of it. His opening sentence was “Translation is the world’s oldest profession.” I was just a young researcher, and wasn’t sure how to put it to him. “Er, Professor, I think that phrase is usually used about a slightly different line of work.” “Ah yes, of course”, he replied. I was eager to discover, a month later, what he would say instead. “Ladies and gentlemen,” he began, “translation is the world’s second oldest profession.”
3. “Sub-” in the mathematical sense, not “inferior”.
4. We refrain here from giving URLs since these are prone to rapid change. We are sure however that most search engines will easily find appropriate web pages using the search term “Machine Translation” or one of the acronyms listed in the text.

References

- Arnold, D., L. Balkan, R. Lee Humphreys, S. Meijer and L. Sadler (1994) *Machine Translation: An Introductory Guide*. Manchester: NCC Blackwell.
- Hutchins, W. J. (1986) *Machine Translation: Past, Present, Future*. Chichester: Ellis Horwood.
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*. London: Academic Press.
- Kay, Martin, Jean Mark Gawron and Peter Norvig (1994) *Verbmobil: A Translation System for face-to-Face Dialog* (CSLI Lecture Notes No. 33). Stanford, CA: Center for the Study of Language and Information.
- Nirenburg, S., H. Somers and Y. Wilks (eds) (2003) *Readings in Machine Translation*. Cambridge, Mass.: MIT Press.
- Trujillo, Arturo (1999) *Translation Engines: Techniques for Machine Translation*. London: Springer Verlag.

CHAPTER 2

The translator's workstation

Harold Somers
UMIST, Manchester, England

1. Introduction

While the original aim of the MT pioneers was fully automatic MT systems, there has also been, at least since the 1966 ALPAC report (see Chapter 1) and possibly even earlier, the view that computers could be used to *help* humans in their translation task rather than to replace them. In this chapter we will look at a range of computer-based tools that have been developed or proposed which can help translators. As we will see, ideas along these lines date back to the 1960s, even when access to computers was not particularly easy to obtain, nor were they especially efficient. In more recent times the ready availability of PCs, as well as the existence and growth of the Internet, could be said to have revolutionised the job of the translator.

The translation activities we will be discussing in this chapter can be broadly classified as **Computer-Aided Translation** (CAT), though often a finer distinction is made between **Machine-Aided Human Translation** (MAHT) and **Human-Aided Machine Translation** (HAMT)¹ implying a distinction between a basically human activity involving computer-based tools on the one hand, and a computer-driven activity requiring the assistance of a human operator. The distinction may be useful at times, though it involves a degree of fuzziness at the edges which should not concern us. Nevertheless, the terminology suggests a spectrum of modes of operation in which the computer plays a progressively bigger part, which can usefully dictate the order of presentation of topics in this chapter.

2. Historical sketch

The idea for a translator's workstation (or “workbench”) is often attributed to Martin Kay, who in 1980 wrote a highly influential memo “The Proper Place of Men and Machines in Language Translation”. However, many of the ideas expressed by Kay had already been hinted at, or even implemented in admittedly crude systems.

In 1966, the ALPAC report — (in)famous for its criticism of attempts at fully automatic MT — recommended among other things the development of computer-based aids for translators. Even before the ALPAC report, the German Federal Armed Forces Translation Agency (the *Bundessprachenamt*) used computers to produce **text-oriented glossaries**, i.e. lists of technical terms and their approved translations based on a given source text. Next came facilities for online access to multilingual **term banks** such as *Eurodicautom* in the CEC and *Termium* in Canada, and programs for terminology management by individual translators. In the late 1970s we also find the first proposal for what is now called **translation memory**, in which previous translations are stored in the computer and retrieved as a function of their similarity to the current text being translated.² As computational linguistic techniques were developed throughout the 1980s, Alan Melby was prominent in proposing the integration of various tools into a translator's workstation at various levels: the first level would be basic word-processing, telecommunications and terminology management tools; the second level would include a degree of automatic dictionary look-up and access to translation memory; and the third would involve more sophisticated translation tools, up to and including fully automatic MT. Into the 1990s and the present day, commercial MT and CAT packages begin to appear on the market, incorporating many of these ideas: an example is shown in Figure 1. And as translators become more computer literate, we see them constructing their own “workstations” as they come to see translation-relevant uses for some of the facilities that are in any case part of the PC.

3. Basic tools

Let us start at the most basic level of computer use by translators. Although probably taken for granted by most translators, **word processing** software is an essential basic computational tool. Modern word processors include many useful facilities such as a **word-count**, a **spell-checker**, a **thesaurus** (in the popular sense of “synonym list”) and — of more dubious use to a translator — grammar and style checkers. Most of these functions are available with most well-known word-processing software packages, though we should note the extent to which all of them are highly language-dependent and language-specific. No problem if we are working into one of the major commercially interesting languages (major European languages — English, French, Spanish, German, Italian, Portuguese, Russian — plus Japanese, Chinese, Korean and to a certain extent Arabic), but simple resources such as those just mentioned may not be available for other “minority” languages (see Chapter 6), or may be of inferior quality. In fact, for some languages such tools may not even be appropriate. For a language which uses a non-alphabetic writing system, like

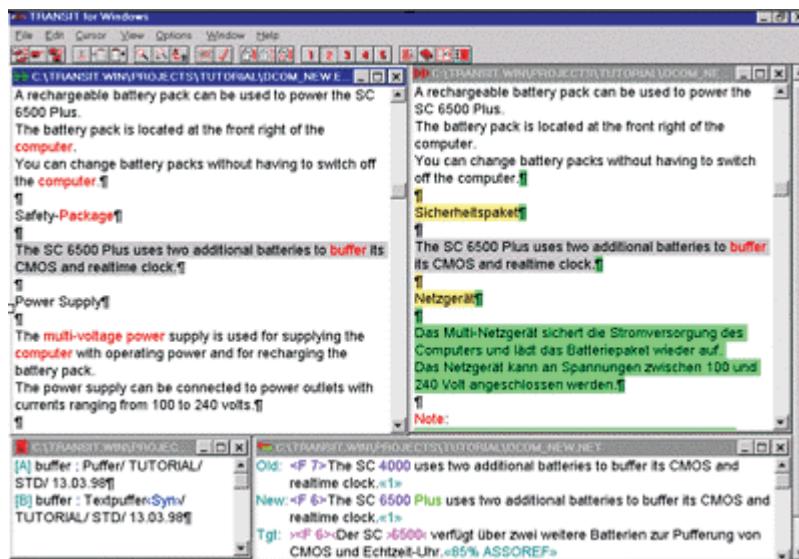


Figure 1. *Transit*: An example of a translator's workstation.

Japanese or Chinese, there isn't really any concept of "spelling" to be corrected by a spell-checker (though of course there are other functions that a word-processor must provide, notably a means of inputting the characters in the first place).

On the subject of writing systems, thankfully much progress has been made in recent years to ensure that the scripts used for most of the world's languages are actually available in word-processors. The Unicode consortium has made efforts to provide a standardised coding for multiple character sets, ensuring unique character codes which enable texts with mixtures of writing systems to be edited and printed and so on. Nevertheless, some problems remain, especially where languages use local variants of a more established writing system (diacritics seem to be a perennial problem), and certainly for many writing systems there is nothing like the range of fonts and type-faces that are available for the Roman alphabet.

Translations need to be revised, and the editing tools that word-processing packages provide are of course very useful. Although not yet commercially available, there has been talk amongst language engineering researchers and developers about the possibility, in the context of a translator's workstation, of **translator-oriented** or **linguistically sophisticated** editing tools: a "translator-friendly" word-processor. Here is envisaged software with the normal word-processing facilities enhanced to facilitate the sort of text editing "moves" that a translator (or, perhaps, a translator working as a post-editor on some MT output) commonly makes. Simple things like transposing two words at the touch of a function key are easy to

imagine, but the software could incorporate more linguistically sophisticated tools such as “grammar-conscious global replace” in which the word-processing software was linguistically aware enough to recognize inflected variants of the word and change them accordingly, for example globally changing *purchase* to *buy* and getting “for free” *purchasing* → *buying* despite the missing *e*, and *purchased* → *bought*. With some “knowledge” of grammar, the word-processor could take care of grammatical consequences of revisions. For example, if you had a text in which the word *fog* had been translated as *brouillard* in French, but you decided *brume* was a better translation, you would have to do more than globally change *brouillard* to *brume*: *brouillard* is masculine, while *brume* is feminine, so some other changes (gender of adjectives and pronouns) may have to be made. You might want to replace *look for* with *seek*, and be hampered by the fact that the word *for* will not necessarily occur right next to the word *look*. The translator-friendly word-processor could also search for “false friends” (e.g. *librairie* as a translation of *library*— see also below) and other “interference” errors, if the user is a competent but not fluent writer of the target language. It might also recognize mixed-language texts and operate the appropriate spell-checker on different portions of text (this paragraph for example contains some French words which my spell-checker is marking as possible misspellings!) Unfortunately, none of these features are as yet found in currently available word-processing software, and it should be clear to the reader that to incorporate them would involve knowledge of grammar and vocabulary, and the ability to analyse the text not unlike that needed to do MT (see Chapter 8).

4. Dictation tools

One technology that is up-and-coming and of interest to many translators is **dictation tools**. As an alternative to typing in their translations, translators are discovering that dictating their draft translation into the computer using **speech recognition** systems can be a great boost to productivity. This gain is due not only to the obvious fact that most people can talk faster than they can type, but to other “hidden” advantages. Michael Benis has suggested that translators are less likely to come out with a clumsy or inelegant construction if they actually have to say it out loud. Typographical errors are also reduced, since dictation software does not insert words that are not found in its dictionary — though of course they may not be the *correct* words, due to the limitations of the software. There is even a gain from the health point of view, since dictation systems allow the translator to get away from the confines of the keyboard, mouse and screen environment responsible for well-documented industrial illnesses.

Dictation systems are not without their drawbacks however. The technology is

still in its infancy, and can make annoying mistakes just because of the inherent difficulties of speech recognition. The user must speak more clearly and slowly than may be natural, and you can expect the system to be confused by homophones (words which sound identical) and even similar-sounding words. Most systems work on the basis of “trigrams”, or sequences of three words, and include extensive statistics on the probability of word sequences. For example, if you dictated the sentences in (1) you could expect the system to get the correct homophone *rode* or *rowed*, because the disambiguating word *bike* or *boat* is within three words. But in a case like (2) it might not get it right.

- (1) I rode the bike. I rowed the boat.
- (2) This {boat, bike} is just like one that I sometimes {rode, rowed}.

Basic errors such as confusing *there* and *their* can nevertheless still be expected, while continuous speech has an unexpected effect on words which might be easy to identify when spoken in isolation. Try reading the following examples aloud to see what the text probably should be.

- (3) It's hard to wreck a nice beach.
- (4) What dime's the neck strain to stop port?

All individuals have slightly different speech patterns, on top of the fact that regional accents can vary hugely. In fact, your own speech can vary significantly from occasion to occasion, for example if you have a cold, are tired, excited and so on. For this reason, dictation systems usually have to be **trained** to recognize the idiosyncrasies in your speech, so that when you first install dictation software, there will be a more or less lengthy training (or “enrolment”) period in which you are asked to read some “scripts” designed to sample a range of speech sounds from which the system can learn your individual phoneme system. In addition, you can “teach” the system additional vocabulary from your own field, by training it on texts of your choice.

Once they have been trained, another way to improve performance is to use the system’s **correction** utility. Many systems include this ability to learn from your corrections so as not to make the same mistake again. If using this feature, it is important to distinguish between correcting *errors*, and changing your mind about what you want to say. If you decided to change *help* to *assist*, you would not want the system to “learn” that the spoken word [help] is written *a-s-s-i-s-t*.

Of course, like most language-related software products, dictation systems are highly language-specific, and as with many such products, you will find a large choice of products for English and the other major European languages, but if you are working into any other languages it may be much harder to find a suitable product.

5. Information technology

A further basic and important element in a translator's workstation comes under the general heading of **information technology**. Many translators nowadays receive and send their work directly in computer-compatible form. Diskettes and writable CDs are excellent media for receiving, sending and storing large amounts of textual — and other — material. Equally, telecommunications are playing an increasing role, whereby translators receive and send material via phone-lines in the form of faxes and e-mail attachments.

What will happen to the translated text is often a concern of the translator, and so **desktop publishing** software might in some sense be part of the translator's workstation. Formatting that needs to be preserved from the source-text can easily be copied over to the target text (in fact translators may simply copy the file and overwrite it with the translation). Text which contains graphics which in turn contain text which must be translated is no longer the printer's nightmare that it might once have been, if the translator has access to the same graphics package as was used to draw the original diagram, and in which text boxes can be simply substituted (Figure 2).

It should be said that the apparent advantages of using this technology can evaporate if for example the source text is badly formatted (with spaces instead of tabs in tables, linefeeds used to force page-breaks, and so on), and some translators may prefer to restrict their work to translating, and leave lay-out and formatting to the experts.

Another recent development in the world of computer-based text handling is the use of **mark-up languages**. The idea is that texts can contain "hidden" markers or **tags** to indicate structural aspects of the text, which in turn can be interpreted for formatting. Users of word-processors will be familiar with the concept of style templates, which are similar: by marking, say, a section header explicitly as such one

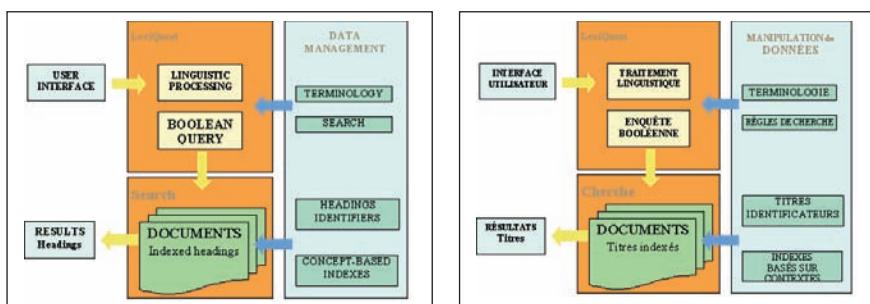


Figure 2. Translating in-figure captions can be easier.

can define separately the formatting associated with the tag, and this can be easily changed for the whole document if necessary. Efforts have been made to standardize the way in which this mark-up is used, and the Standard Generalized Mark-up Language SGML is widely used. If you look at the “page source” of a web page, you will see HTML, which is very similar: tags are seen as symbols within angle brackets, and generally come in pairs with the “closing” symbol the same as the “opening” symbol but preceded by a slash. For example, a level-3 heading might be indicated <h3>thus</h3>. While SGML is widely used to define **document structure** and with its formatting conventions, it can also be used for a wide variety of other purposes, including annotating texts in many ways relevant to the translation process, for example, inserting codes identifying technical terms (and their translations), indicating grammatical information on ambiguous words, identifying the source of the translation of each sentence (human, MT system, translation memory, etc.), and any other commentary that the author or translator might wish to add, such as instructions to the printer, but which will not appear in the final document.

6. Lexical resources

Beyond word-processing and related tools, the translator's workstation should facilitate access to an array of **lexical** resources, in particular online dictionaries and term banks.

Online dictionaries may take the form of computer-accessible versions of traditional printed dictionaries (Figure 3), or may be specifically designed to work with other applications within the workstation. The online dictionaries may of course be mono-, bi- or multilingual. The way in which information associated with each entry in the dictionary is presented may be under the control of the user: for example, the translator may or may not be interested in etymological information, pronunciation, examples of usage, related terms and so on. Online dictionaries can be little more than an on-screen version of the printed text; or else they may take advantage of the flexible structure that a computer affords, with a hypertext format and flexible hierarchical structure, allowing the user to explore the resource at will via links to related entries. For example, clicking on the highlighted word *passage* in Figure 3 brings up the screen shown in Figure 4.

The user may or not be allowed to edit the contents of the dictionary, adding and deleting information including entire entries. Where the online dictionary is also used to provide a draft translation (see below), it is normal to allow the user to add entries. For example, Figure 5 shows the term *work station* being added to a dictionary entry for *work*.

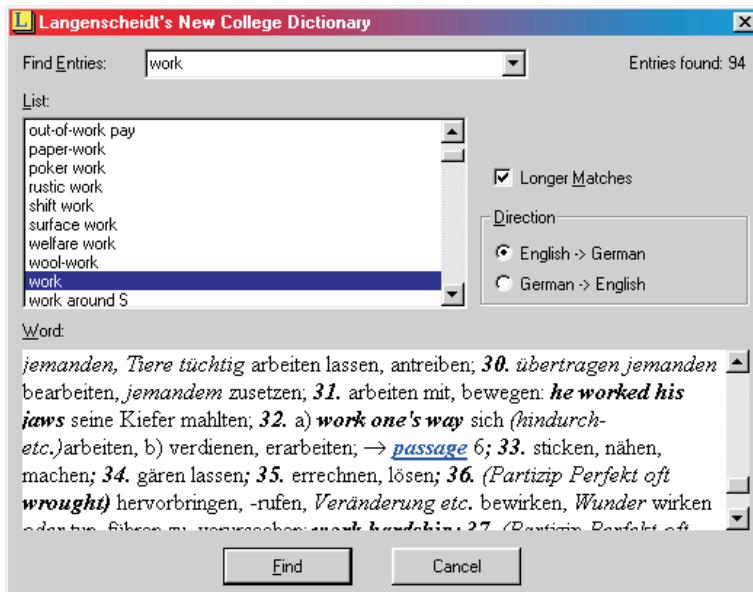


Figure 3. Online version of Langenscheidt's *New College Dictionary* (from the *T1 Professional* system).

An important resource for translators is of course technical terminology. Online access to **term banks** was one of the earliest envisaged CAT tools, and with the growth of the Internet the focus nowadays is on licensed access to centrally maintained terminology rather than local copies, although there is obviously also a place in the translator's workstation to allow translators to maintain their own lists of terminology in a variety of formats. See Chapter 4 for more discussion of terminology and the translator.

It would be narrow-minded to assume that the only sources of information that a translator needs are collections of words: easy access to a wide range of other types of information can be part of the translator's workstation: a gazetteer can be useful to check proper names, as can a list of company names. Encyclopedias and other general reference works are all useful resources for the translator, and all can be integrated into the translator's workstation. Of course, as many translators already know, resources such as these, and many more, are readily available on the World Wide Web, access to which would be an essential element of the translator's workstation.

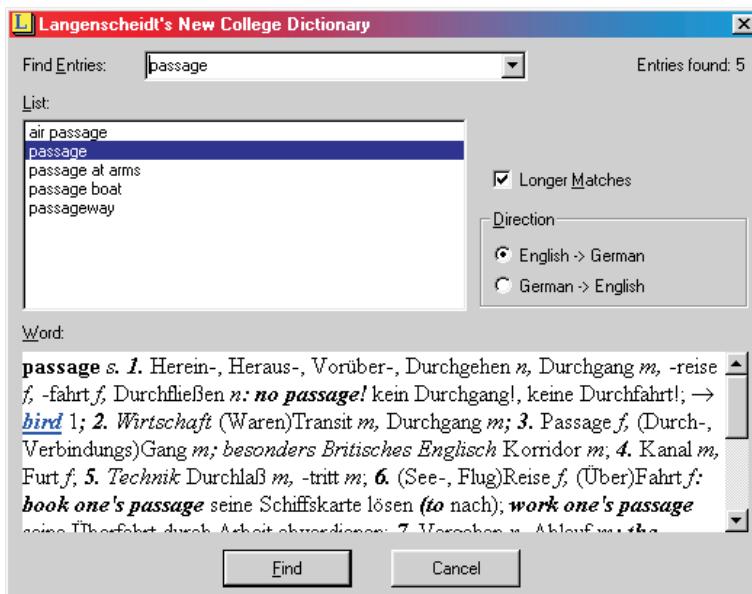


Figure 4. Dictionary entry shown by clicking on link in Figure 3.

7. Features of typical commercial MT systems

Software with some translation capability will be an integral part of the translator's workstation. The most important feature of this is that it is under the user's control. In this section we will look at the typical commercial MT system and consider to what extent it can be used by a translator.

The first thing to notice is that, rightly or wrongly, commercial MT systems are designed primarily with use by non-linguists in mind. This is evident in the packaging, and in the wording of user manuals, like the following, regarding updating dictionary entries:

Many users, still haunted by memories of grammar classes at school, will be daunted by this idea. But with T1 there's really no need to worry. Special user-friendly interfaces permit you to work in the lexicon *with a minimum of knowledge and effort.* (Langenscheidt's *T1 Professional*, User Manual, p. 102; emphasis added)

So, the question quickly arises: Are these systems useful for real translators? Individuals should experiment with the less expensive systems (though bear in mind that cost and quality go hand in hand, and the very basic systems can easily give a

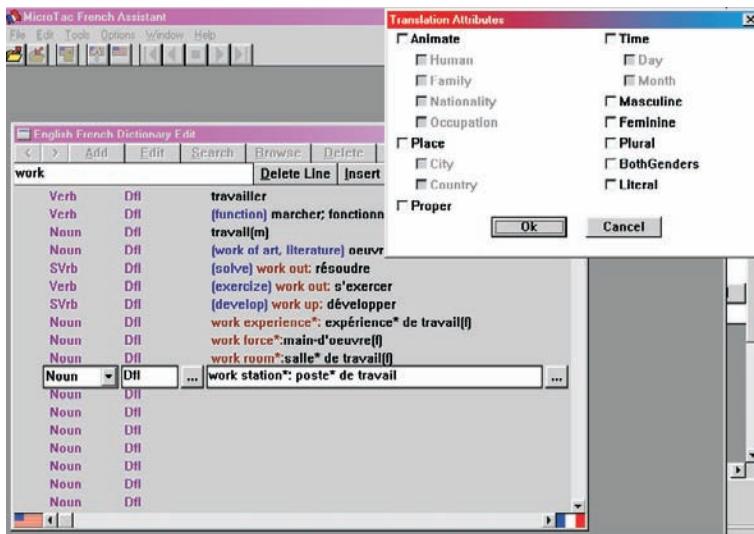


Figure 5. Adding to a dictionary entry (from the *French Assistant* system).

bad impression of the slightly more expensive ones). Let us consider what you are likely to get from an MT system, and how you might put it to use.

The typical system presents itself as an extended word-processing system, with additional menus and toolbars for the translation-related functions (Figure 6) including translation memory which we deal with separately in the next chapter.

Often, the suggested set-up has the source text shown in one window, with a second window for target text, in which the source text is initially displayed, to be over-written by the translation (Figure 7). Often, the user can customize the arrangement, for example to have source and target text side-by-side rather than vertically arranged, as shown. Figure 7 also shows how the formatting is maintained between the two windows.

In its most simple mode of use, the user highlights a portion of text to be translated, as seen in Figure 7. The draft translation is then pasted in the appropriate

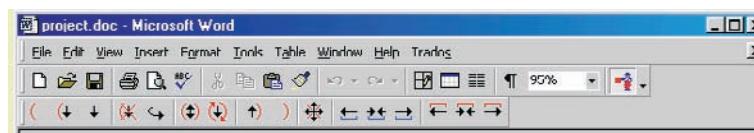


Figure 6. Word-processor with additional menus and toolbars (from the Trados system).

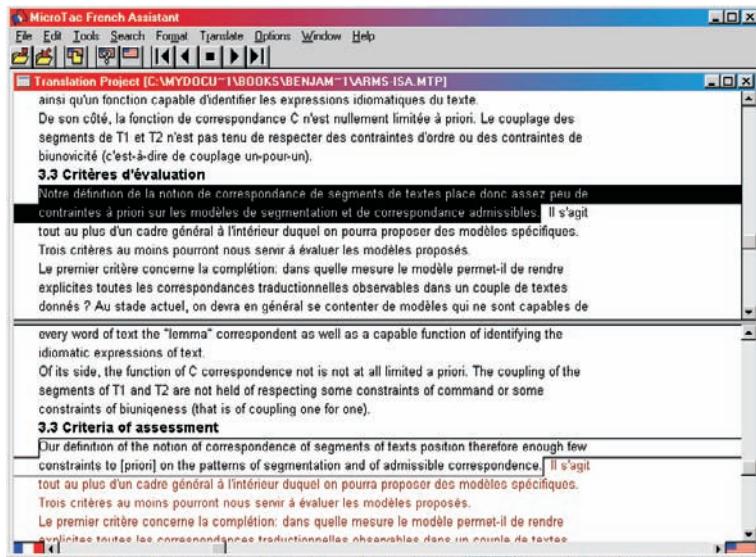


Figure 7. Source and target text in parallel windows (from *French Assistant*).

place in the target text window, ready for **post-editing**. Allowing the user to determine which portions of text should be sent to the MT system gives the user much more control over the process (although some systems will try to translate a whole sentence regardless of what text has been highlighted). If the user really can determine what text is to be translated, they will quickly learn to assess what types of text are likely to be translated well, and can develop a way of working with the system, translating more difficult sections immediately “by hand”, while allowing the system to translate the more straightforward parts.

More sophisticated modes of operation are also possible. Most CAT systems allow the user to run a “new word” check on the source text, and then to update the system’s dictionary using the list of “unknown” words. Many systems offer a choice of **interactive** translation in which the system stops to ask the user to make choices (Figure 8). Many CAT users however have suggested that this slows down the process, since the system repeatedly offers the same choices, asks “stupid” questions, and apparently never “remembers” a relevant choice made earlier in the translation of the same text (though to do so correctly would actually require some quite sophisticated software design).

Full word-processing facilities are of course available in the target-text window to facilitate post-editing. With many systems, the same is true of the source-text window, which simplifies the task of **pre-editing**, i.e. altering the source-text so as to

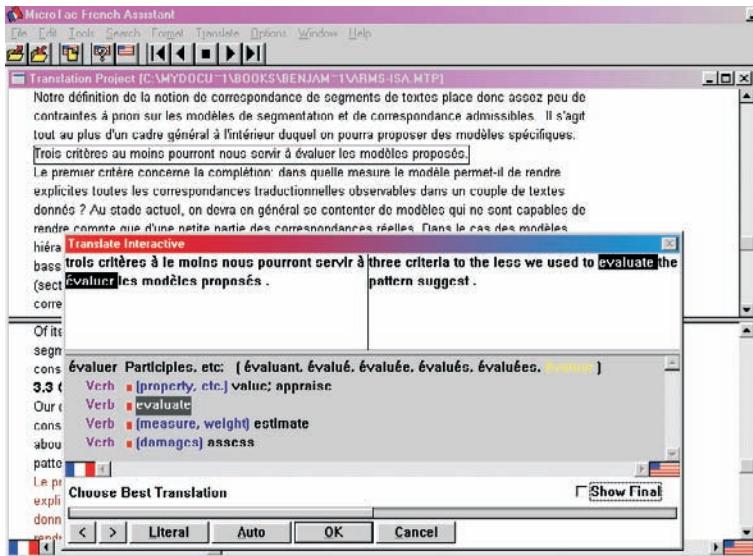


Figure 8. Interactive translation (*French Assistant*).

give the MT system a chance of doing a better draft translation. The juxtaposition of the two windows, and the ease of sentence-by-sentence translation suggests a novel method of trial-and-error computer-aided translation which has been called “post-editing the source text”. The idea behind this apparently counterintuitive activity is that the user can see what kind of errors the MT system makes, and can then change the source text in response to these errors. Post-editing the source rather than the target text might involve the user in less work. For example, suppose we have a text containing a recipe in French: many of the sentences are instructions, expressed in the French infinitive form, as in (5).

- (5) *Peler les pêches. Dénoyer. Couper les fruits en quartiers.*
to-peel the peaches. to-stone. to-cut the fruits into quarters

Now let us suppose that, unfortunately, our MT system always seems to translate infinitives in this type of sentence using the *-ing* form of the verb, instead of the imperative (6).

- (6) Peeling the peaches. Stoning. Cutting the fruit into quarters.

Apart from the form of the verb, the translation is usable. But note, assuming that this error is repeated throughout a reasonably long text, the post-editing effort involved in correcting it. A simple search-and-replace deleting *-ing* will not work,

because that would leave forms like **Ston* and **Cutt*. An alternative is to edit the source-text, changing the infinitives ending in *-erto* to imperatives ending in *-ez*. With a few exceptions this can be achieved by a (careful) global search-and-replace. Although it renders the source text less elegant (and one can give examples of similar fixes that actually make the source text ungrammatical), this does not matter, since the text in the source window can simply be a working copy of the original, and no one need see the cannibalised version.

8. Other corpus-based resources

A major interest of computational linguistics in recent years has been “corpus linguistics”. A **corpus** is a collection of text, usually stored in a computer-readable format. The example database of a translation memory (see Chapter 3) is an example of a corpus, with the particularly interesting property of being an **aligned parallel corpus**, by which is meant that it represents texts which are translations of each other (“parallel”), and, crucially, the corpus has been subdivided into smaller fragments which correspond to each other (hence “aligned”). This kind of corpus is an extremely useful resource for translators, and a number of tools can be built which make use of it.

One of the most useful is the **concordance**, also sometimes known as a keyword in context (KWIC) list: it is a tool that literature scholars have used for many

```

1 hed it off. * * * 'What a curious feeling!' said Alice; 'I must b
1 against herself, for this curious child was very fond of pretendi
2 'Curiouser and curiouser!' cried Alice (she was so muc
2 Eaglet, and several other curious creatures. Alice led the way,
4 -- and yet - it's rather curious, you know, this sort of life!
6 eir heads. She felt very curious to know what it was all about,
6 out a cat! It's the most curious thing I ever saw in my life!' S
7 ht into it. 'That's very curious!' she thought. 'But everything'
7 hought. 'But everything's curious today. I think I may as well g
8 Alice thought this a very curious thing, and she went nearer to w
8 she had never seen such a curious croquet-ground in her life; it
8 seen, when she noticed a curious appearance in the air: it puzz
9 next, and so on.' 'What a curious plan!' exclaimed Alice. 'That's
10 : 'and I do so like that curious song about the whiting!' 'Oh,
10 th, and said 'That's very curious.' 'It's all about as curious a
10 ous.' 'It's all about as curious as it can be,' said the Gryphon
11 moment Alice felt a very curious sensation, which puzzled her a
11 er the list, feeling very curious to see what the next witness wo
12 ad!' 'Oh, I've had such a curious dream!' said Alice, and she tol
12 her, and said, 'It was a curious dream, dear, certainly: but no

```

Figure 9. Concordance of the word *curious* in *Alice's Adventures in Wonderland*.

years. This alternative name gives a clue as to what a concordance is, namely a list of occurrences of a given word, showing their context. Figure 9 shows an example of this, a list of all the occurrences of the word *curious* (or more accurately, the sequence of characters *c-u-r-i-o-u-s*) in Lewis Carroll's famous book, *Alice's Adventures in Wonderland*.

A listing such as this is of interest in itself since it shows the range of use of an individual word. For a translator, of more interest is a **bilingual concordance**, in which each line is linked to the corresponding translation. This enables the translator to see how a particular word (Figure 10) — or more usefully a phrase or a technical term (Figure 11) — has been translated before.

Apart from the obvious use as a source of suggestions for the translator, the bilingual concordance can also be used for a number of other purposes. Once a (large) text has been translated, if it is saved as an aligned bilingual corpus, the tool can be used to search for a pair of terms, and in this way check for possible “false friends”: Figure 12 shows examples where *librairie* has been translated as *library*, whereas *bookshop* is the more usual translation. In a similar way, consistency of translation can be checked. Figure 13 for example shows that the verb *rise* is usually translated as *prendre la parole*, but is occasionally translated as *intervenir*, which may or may not be considered “incorrect”. The bilingual concordance also shows (example 6) that even in parliamentary language a technical term can also be used in its everyday sense.



Figure 10. An English–Japanese bilingual concordance listing for the word *Translator's* (Trados).

Document Collection: Canadian Hansard (1986-1993)	
Expression: point of order	
1. Monsieur le Président, j'invoque le Règlement.	“Mr. Speaker, point of order ”.
2. À l'époque, on pouvait invoquer le Règlement même pendant la période des questions.	In those days one could interrupt anything with a point of order , even Question Period.
3. Monsieur le Président, j'invoque le Règlement.	Mr. Speaker, I rise on a point of order .
4. J'invoque le Règlement, monsieur le Président.	Mr. Speaker, I rise on a point of order .
5. Monsieur le Président, j'invoque le Règlement.	Mr. Speaker, I rise on a point of order .
6. Monsieur le Président, j'invoque le Règlement.	Mr. Speaker, I rise on a point of order .
7. Avant de passer la parole au député de Moncton, je passe la parole au secrétaire parlementaire du leader du gouvernement à la Chambre pour un rappel au Règlement.	Before I recognize the hon. member for Moncton I have a point of order from the hon. parliamentary secretary to the government House leader.
8. Pour un rappel au Règlement, monsieur le Président.	I rise on a point of order , Mr. Speaker.
9. Monsieur le Président, j'invoque le Règlement.	Mr. Speaker, a point of order .
10. J'invoque le Règlement, monsieur le Président, pour déclarer que ce que vient de dire le député est totalement et absolument faux.	On a point of order , Mr. Speaker. I want to say that the point the member just made is absolutely and totally false.

Figure 11. Bilingual concordance of the phrase *point of order* in the Canadian Hansard.³

Document Collection: Canadian Hansard (1986-1993)	
English Expression: library	
French Expression: librairie	
1. Pas de librairie, pas de cinéma, pas de salle d'exposition, pas de piscine, pas de musée des beaux-arts, pas de vélodrome, pas de métro, etc.	No library, no movie theatre, no exhibition hall, no swimming pool, no art gallery, no velodrome, no subway, and so forth.
2. J'ai demandé à la librairie du Parlement d'examiner tous les documents--prévisions budgétaires, dépenses réelles, documents du Conseil du Trésor et ainsi de suite.	I asked the Parliamentary Library to go through all the documents--the Estimates, the actual expenditures, Treasury Board documents, and so on.

Figure 12. Bilingual concordance of the word-pair *librairie–library* in the Canadian Hansard.

<p>Document Collection: Canadian Hansard (1986-1993) Expression: rise</p>	
<p>1. Madame la Présidente, j'aurai l'honneur de prendre la parole aujourd'hui pour féliciter le gouvernement fédéral d'avoir créé ce printemps un groupe de travail chargé d'examiner les mesures nécessaires pour améliorer sa politique d'aide aux magazines canadiens.</p>	<p>Madam Speaker, I rise today to applaud the initiative of the federal government in the establishment this spring of a task force to review necessary measures to enhance its policy in support of the Canadian magazine industry.</p>
<p>2. Madame la Présidente, je prends la parole aujourd'hui pour rendre hommage à quelqu'un de très spécial au sein de notre parti et de la Chambre.</p>	<p>I rise today to pay tribute to a very special person within our caucus, our party and this honoured place.</p>
<p>3. Madame la Présidente, en ce jour historique, je veux prendre la parole à la Chambre pour remercier mes collègues et les personnes qui m'ont appuyée de m'avoir permis de devenir la première femme qui sera assermentée, le 25 juin prochain, à titre de première ministre du Canada.</p>	<p>Madam Speaker, I rise in the House today with a great sense of history to thank my colleagues and my supporters for providing me with the opportunity to be the first woman who will be sworn in as the Prime Minister of Canada on June 25.</p>
<p>4. Madame la Présidente, je suis heureux de prendre la parole aujourd'hui pour rendre hommage à l'un de nos collègues les plus distingués, notre ami, le député de Vancouver-Sud et Président de la Chambre, l'honorable John Fraser.</p>	<p>Madam Speaker, today it is my pleasure to rise to pay tribute to one of our most distinguished colleagues, one of the most distinguished members of the House, our friend and colleague, the hon. member for Vancouver South, the Speaker of the Chamber, the Hon. John Fraser.</p>
<p>5. Madame la Présidente, c'est vraiment un honneur que de prendre la parole aujourd'hui, au nom de mes collègues du caucus néo-démocrate, pour rendre hommage à une personne qui a certainement été un des parlementaires et présidents les plus remarquables que ce pays ait connu.</p>	<p>Madam Speaker, it is indeed an honour to rise today on behalf of my colleagues in the New Democratic caucus to pay tribute to one of the most outstanding parliamentarians and speakers this country has witnessed.</p>
<p>6. Je suis heureux d'avoir pu quitter le cabinet de mon dentiste qui vient tout juste de m'extraire une dent.</p>	<p>I am pleased to be able to rise from the dentist's chair where a few moments ago I had a tooth jerked out.</p>
<p>7. Madame la Présidente, on a présenté les hommages et dit tout ce qui s'imposait, mais je tenais à me lever pour faire l'éloge de notre Président.</p>	<p>Madam Speaker, everything has pretty well been said, but I felt I wanted to rise to pay tribute to our Speaker.</p>
<p>8. Monsieur le Président, j'invoque le Règlement.</p>	<p>Mr. Speaker, I rise on a point of order.</p>

Figure 13. Bilingual concordance of the word *rise* in the Canadian Hansard.

9. Conclusion

In this chapter we have seen that the translator's workstation represents the most cost-effective facility for the professional translator, particularly in large organisations. It makes available to the translator at one terminal (whether at an individual computer or as part of a company network) a range of integrated facilities: multilingual word processing, electronic transmission and receipt of documents, spelling and grammar checkers (and perhaps style checkers or drafting aids), publication software, terminology management, text concordancing software, access to local or

remote term banks (or other resources), translation memory (for access to individual or corporate translations), and access to automatic translation software to give rough drafts. The combination of computer aids enables translators to have under their own control the production of high quality translations.

Further reading

Hutchins (1998) gives a detailed history of the development of the translator's workstation. Isabelle and Church (1997) is a collection of proposals for translator's tools, though some of the papers are rather technical. Kay's "Proper Place" paper is reprinted in this collection, along with a number of peer-group commentaries. Alan Melby has written several articles outlining his ideas for a translator's workstation. His contribution to Newton (1992) may be the most easily accessible, and this volume contains a number of other essays of interest. Cormier and Estival (1992) is another collection of articles, mostly in French, while Kugler et al. (1995) also contains numerous interesting contributions. Bowker's (2002) recent book is highly recommended.

Concentrating on individual issues, O'Hagan (1996) and Ashworth and O'Hagan (2002) have some interesting views on the role of telecommunications in translation. The idea of post-editing the source text is explored in Somers (1997). Various corpus-based translation tools have been developed by RALI (Laboratoire de Recherche Appliquée en Linguistique Informatique, Université de Montréal), and articles describing them, as well as online demos, are available at the group's website www-rali.iro.umontreal.ca/Accueil.en.html. On the topic of dictation systems, Benis (1999) gives an excellent overview together with a review of several systems. Samuelsson-Brown (1996) discusses this and some other technology-based translator's aids.

Notes

1. There is no doubt some logic and reason for the preference of the term "machine" over "computer", though what it may be seems shrouded in the mists of time.
2. Because of the importance and interest in this tool, we devote a separate chapter to translation memory systems.
3. Figures 10, 11 and 12 are based on the *TransSearch* system developed by RALI, Université de Montréal.

References

- Ashworth, David and Minako O'Hagan (2002) *Translation-mediated Communication in a Digital World: Facing the Challenges of Globalization and Localization*. Clevedon, England: Multilingual Matters.
- Benis, Michael (1999) "It's Good to Talk", originally published in *ITI Bulletin*. Available at transref.org/default.asp?docsref=/u-articles/Benis1.asp.
- Bowker, Lynn (2002) *Computer-aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.
- Cormier, Monique C. and Dominique Estival (1992) "Études et recherches en traductique: Studies and researches [sic] in machine translation", Numéro spécial, *Meta* 37.4.
- Hutchins, John (1998) "The Origins of the Translator's Workstation", *Machine Translation* 13, 287–307.
- Isabelle, Pierre and Kenneth W. Church (eds) (1997) "New Tools for Human Translators", Special Issue, *Machine Translation* 12.1–2.
- Kugler, M., K. Ahmad and G. Thurmair (eds) (1991) *Translator's Workbench: Tools and Terminology for Translation and Text Processing*. Berlin: Springer.
- Newton, John (ed.) (1992) *Computers in Translation: A Practical Appraisal*. London: Routledge.
- O'Hagan, Minako (1996) *The Coming Industry of Teletranslation: Overcoming Communication Barriers through Telecommunication*. Clevedon, England: Multilingual Matters Ltd.
- Samuelsson-Brown, Geoffrey (1996) "New Technology for Translators", in Rachel Owens (ed.) *The Translator's Handbook*, 3rd edition, London: Aslib, pp. 279–293.
- Somers, Harold (1997) "A Practical Approach to Using Machine Translation Software: 'Post-editing' the Source Text", *The Translator* 3, 193–212.

CHAPTER 3

Translation memory systems

Harold Somers
UMIST, Manchester, England

1. Introduction

One of the most significant computer-based aids for translators is the now widely used **translation memory** (TM).¹ First proposed in the 1970s, but not generally available until the mid 1990s, the idea is that the translator can consult a database of previous translations, usually on a sentence-by-sentence basis, looking for anything similar enough to the current sentence to be translated, and can then use the retrieved example as a model. If an exact match is found, it can be simply cut and pasted into the target text. Otherwise, the translator can use it as a suggestion for how the new sentence should be translated. The TM will highlight the parts of the example(s) that differ from the given sentence, but it is up to the translator to decide which parts of the target text need to be changed. Figures 1 and 2 show how different systems present this information.

The key to the process is efficient storage of the sentences in the database, and, most importantly, an efficient matching scheme. In this chapter we will look at the different ways that a TM's database can be built up, and at how the matching

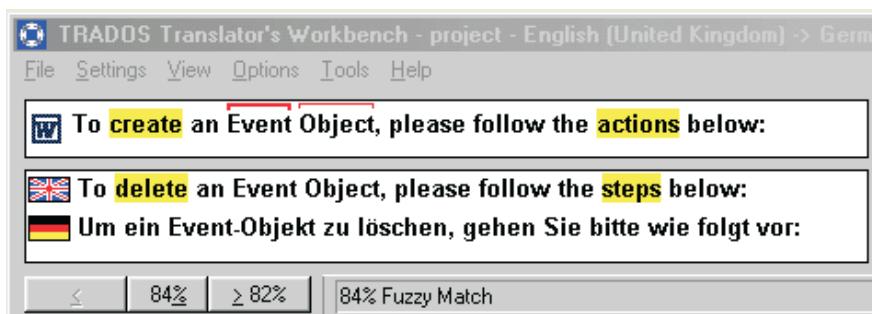


Figure 1. *Trados's* translation memory window showing partial match.

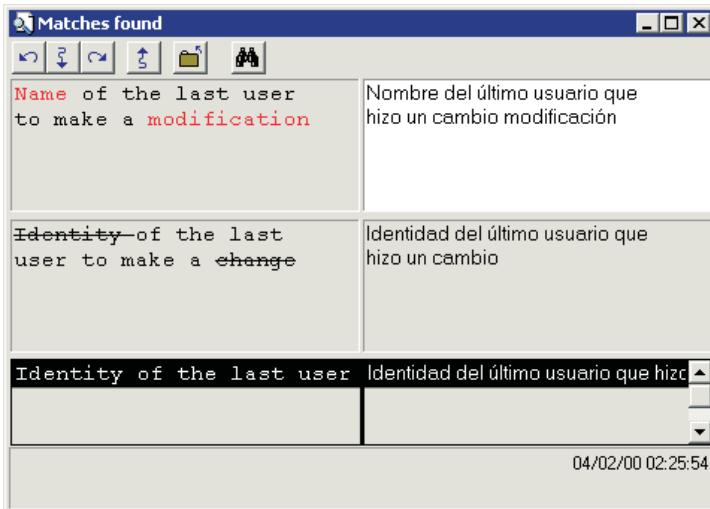


Figure 2. A similar feature in Atril's *Déjà Vu* system.

function works. We give some suggestions about how a TM system can be evaluated, and we end the chapter with a brief discussion of the related technique of Example-based MT.

2. Historical sketch

The original idea for TM is usually attributed to Martin Kay who, as long ago as 1980, wrote a highly influential paper entitled “The Proper Place of Men and Machines in Language Translation” in which he proposed a basic blueprint for what we now call translator’s workstations. In fact, the details relating to TMs are only hinted at obliquely:

... the translator might start by issuing a command causing the system to display anything in the store that might be relevant to [the text to be translated] Before going on, he can examine past and future fragments of text that contain similar material. (Kay, 1980: 19)

Interestingly, Kay was pessimistic about any of his ideas ever actually being implemented. But Kay’s observations are predated by the suggestion by EEC translator Peter Arthern, that translators could benefit from on-line access to similar, already translated documents, and his proposals quite clearly describe what we now call TMs:

It must in fact be possible to produce a programme [sic] which would enable the word processor to ‘remember’ whether any part of a new text typed into it had already been translated, and to fetch this part, together with the translation which had already been translated, Any new text would be typed into a word processing station, and as it was being typed, the system would check this text against the earlier texts stored in its memory, together with its translation. In effect, we should be operating an electronic ‘cut and stick’ process which would, according to my calculations, save at least 15 per cent of the time which translators now employ in effectively producing translations. (Arthern, 1981: 318).

Alan Melby (1995: 225f) suggests that the idea might have originated with his group at Brigham Young University (BYU) in the 1970s. What is certain is that the idea was incorporated, in a very limited way, from about 1981 in *Alps*, one of the first commercially available MT systems, developed by personnel from BYU. This admittedly rather limited tool was called “Repetitions Processing”. The much more inventive name of “translation memory” does not seem to have come into use until much later.

While a small number of research papers on the subject appeared in the late 1980s, it was not until the mid 1990s that TM systems became commercially available, and then in a short period of time they were quickly accepted by users, and several companies released competing products.

3. Building the database

A prerequisite for a TM system is of course a database of translation examples. Known to computational linguists as an “aligned parallel corpus”, there are principally three ways of building a TM database: building it up as you go along, importing it from elsewhere, or creating it from a parallel text.

3.1 Building it as you go

Perhaps the simplest method is to build it up as you go along. Each sentence you translate is added to the database. Obviously, if you are working on a text that is similar to one you worked on before, you can load up the database that you created last time and continue to add to it this time. Conversely, if you are working on different projects and want to develop separate databases for each of them, this can also be done. Unfortunately, this method of developing the database is painfully slow, and there will be a long lead time before the translator really feels the benefit of the software.

3.2 Importing someone else's

The next simplest method is to “import” the database from elsewhere. With the proliferation of TM products, and the increasing numbers of translators using them, it makes sense for users to share their assets. Fortunately, and despite the variety of software products, developers have agreed a common interchange format which means that TM databases developed using one product can be “imported” into another. This is thanks to the TMX (Translation Memory eXchange) agreement brokered by OSCAR (Open Standards for Container/Content Allowing Reuse), a special-interest group within LISA, the Localisation Industry Standards Association. The significance of this should not be overlooked. TM databases are not simply text files. In order for the matching algorithms to work efficiently (see next section), the databases have to be highly structured, with indexes to facilitate efficient retrieval of examples. Many TM systems also feature a terminology matching facility, or other add-ons. In particular, it is often the case that as items are added to the database they can be annotated with additional information such as their source, date, validation code, the name of the translator; and as they get used, some systems maintain statistics which can influence the matching algorithm so that it chooses more frequently used examples wherever possible. On top of this there is the question of compatibility of different word-processing formats. All of these elements and more are subject to TMX agreements.

3.3 Aligning a parallel text

The third, and technically most complex, alternative is to take an existing translation together with the original text and have the software build a TM database from it automatically. This involves **alignment** above all else, though as the previous paragraph indicated, once aligned there will be an amount of indexing and other database manipulations that need not concern us here.

Alignment involves matching up the source text and the translation segment by segment into translation pairs. “Segments” are usually understood to correspond to sentences or other more or less easily distinguishable text portions, such as titles. If the translation is straightforward, then so is the alignment. But three factors can make alignment more difficult than it at first seems: one is the difficulty of accurately recognizing where sentences begin and end; the second is the fact that — depending on the language pair — a single sentence in one language may not necessarily correspond to a single sentence in the other language; the third factor is that translators may more or less freely change the relative order of sentences in the translation.

We can illustrate the first point rather easily. A simple practical definition of a

sentence (ignoring grammatical norms about the need to include a finite verb) might be “a sequence of characters ending in a full stop”. But the examples below show how simplistic this definition might be.

- (1) Chapter Five
- (2) Dr. Smith met his cousin, recently arrived from the U. S. A., at St. Pancras Station.
- (3) What is the meaning of life? Forty-two!

The second problem is more or less widespread depending on the language-pair and text-type concerned. The following example, from parallel versions of the online *Frankfurter Allgemeine Zeitung* show how addressing different audiences can lead to significant differences. The German sentence in (4a), for which we provide a close translation in (4b), appeared as (4c) in the English version.

- (4) a. *Der Sicherheitsberater des Bundeskanzlers, Steiner, äußerte sich ähnlich, wenn er die Vertreter der Bundestagsfraktionen unterrichtete. Es lasse sich nicht absehen, wann das Bundeskabinett und dann der Bundestag über einen Entsendebeschluß zu befinden hätten.*
- b. The Chancellor's security advisor, Steiner, expressed the same opinion when he briefed representatives of the Bundestag parliamentary groups. It could not be predicted when the cabinet and then the Bundestag would come to a decision on dispatch.
- c. Chancellor Gerhard Schröder's foreign policy advisor, Michael Steiner, briefed representatives of the Bundestag parliamentary groups telling them there was no timetable for the cabinet and, later, the Bundestag, to make a decision on dispatching troops.

It could be argued that these texts are so different that to treat them as parallel, and therefore load them into a TM's database, would be misguided. But it is only on close inspection of the two texts, which are superficially quite apparently parallel, that the differences between the texts appear.

The problem is exacerbated with less closely related language pairs like English and Chinese, Japanese, Arabic and so on, where, hand in hand with a different writing system, we find non-corresponding punctuation systems and a quite different notion of “sentence”. In example (5), from Gerber and Hovy's (1998) paper on the subject, we show only a literal translation from Japanese, and preferred English translation which splits the single Japanese sentence into three.

- (5) a. On the one hand, concerning long-distance travel, as for the present subsonic planes, because the Tokyo–New York flight time for example can be as much as 12 hours, (and) the demand for shorten-

ing of the long-distance flight time is great, development opportunities for new projects including developing supersonic aircraft starting in the 21st century are increasing.

- b. On the one hand, considering long-distance travel with the present subsonic planes, there is a great demand for shortening long-distance flight time. For example, the Tokyo–New York flight time can be as much as 12 hours. Because of this, development opportunities for new projects including developing supersonic aircraft starting in the 21st century are increasing.

Because of this, many TM programs offer more or less sophisticated alignment tools which make a first attempt at alignment, but allow the user to correct the alignments proposed. Figure 3 shows an example of such a tool: the central panel shows the proposed alignments, the thickness of the line being some indication of the system's confidence in its results. The tool allows the user to position the cursor over the little boxes and redo the alignment like they were using a graphics tool.

In the top window in Figure 3 we see that the system has also performed a “structure-level” alignment. It is not unusual to find texts where the order of larger sections differs, so it is useful to be able to adjust the alignment at this higher level. Even within segments, we can also find sentences whose order has been reversed in

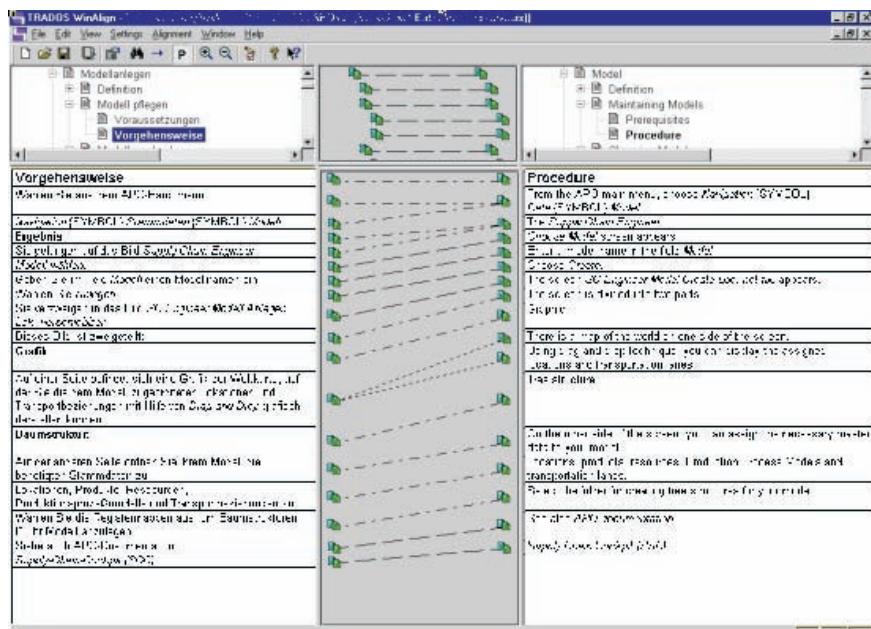


Figure 3. Output of an alignment tool.²

the translation, leading to a “crossing” alignment. These are difficult to spot automatically, especially if the segments are of a similar length.

This is because of the way these alignment tools actually work. For obvious reasons, details of commercial products are sometimes difficult to obtain, but what is almost certain is that most alignment tools work on little more than the crude though reasonable assumption that long sentences give rise to long translations, and short sentences to short translations. There is an extensive literature on alignment techniques, which mentions various means by which alignments based on this simple assumption can be improved, but it is unlikely that many of the more sophisticated (and accordingly time-consuming) methods are incorporated in commercial tools. For example, alignment can certainly be improved by looking for “cognates” — a slightly misleading term since it refers to any words which are significantly similar in the source and target text, irrespective of whether they are historically related in the philological sense of the word “cognate”, and applies mostly to literal strings like proper names, dates, numbers and so on. The point is that these provide good “anchor points” for the alignment. This is likely to be the limit of “linguistic” information used by alignment programs, one reason being that it is important that they be as neutral as possible with regard to language-pair: the same alignment procedure can apply to any pair of texts, regardless of the languages concerned. This need for language-pair independence will crop up again in the next section.

4. Matching

Obviously the most important function for a TM system is its ability to match the sentence to be translated against the database. Where there is an exact match, the system will normally take the corresponding target-language phrase and paste it directly into the target text, though the user will always have the option of rejecting it. Where there is not an exact match, the system presents one or more close matches, with the differences highlighted (see Figures 1 and 2). For example, if (6) is the sentence to be translated, and the database contains (7a) with its accompanying translation (7b), the system can highlight the differences between (6) and (7a), as we do here, perhaps using different colours to indicate deletions, insertions and substitutions. Notice however that it is unable to identify which words in (7b) have to be changed: this task remains in the translator’s hands.

- (6) The large tray can hold up to four hundred sheets of A4 paper.
- (7) a. The small paper tray ... holds up to three hundred sheets of A5 paper.
 - b. *Die kleine Papierkassette fasst bis zu dreihundert Blatt in A5-Format.*

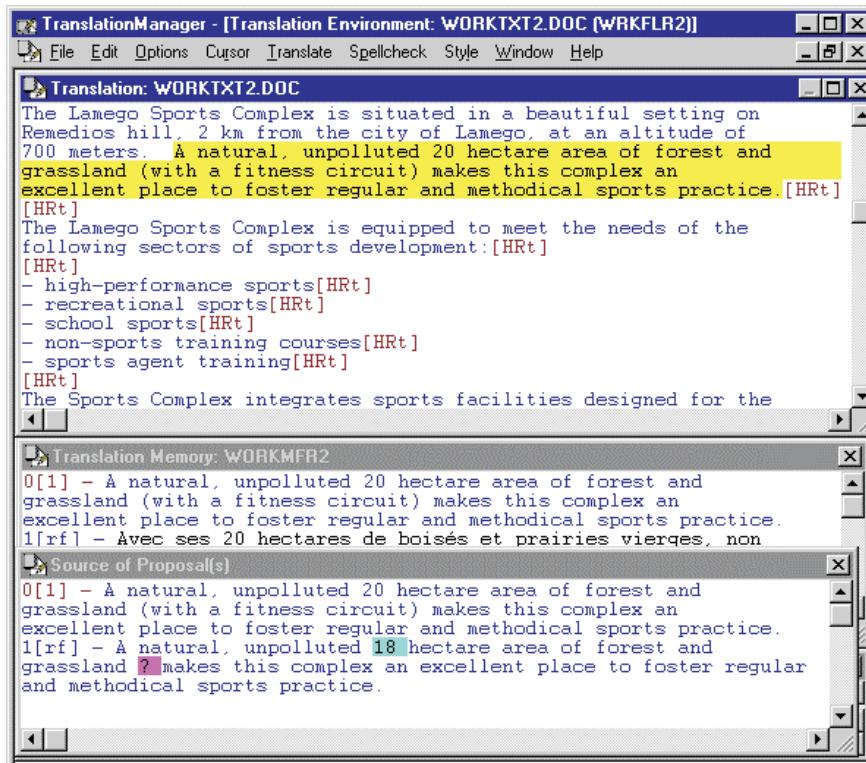


Figure 4. IBM's *Translation Manager* showing multiple matches.

Some systems permit the user to view several matches simultaneously, arranged in order of “fuzziness” (Figure 4).

4.1 Fuzzy match score

Most current commercial TM systems offer a quantitative evaluation of the match in the form of a “score”, often expressed as a percentage, and sometimes called a “fuzzy match score” or similar. How this score is arrived at can be quite complex, and is not usually made explicit in commercial systems, for proprietary reasons. In all systems, matching is essentially based on character-string similarity, but many systems allow the user to indicate weightings for other factors, such as the source of the example, formatting differences, and even significance of certain words.

The character-string similarity calculation uses the well-established concept of “sequence comparison”, also known as the “string-edit distance” because of its use

in spell-checkers, or more formally the “Levenshtein distance” after the Russian mathematician who discovered the most efficient way to calculate it.³ The string-edit distance is a measure of the minimum number of insertions, deletions and substitutions needed to change one sequence of letters into another. For example, to change *waiter* into *waitress* requires one deletion and three insertions (there are two ways to do this—either delete the *e* and add *ess* or insert an *r* after the *t*, delete the superfluous *r* and add *ss*—but either way the score is the same). The measure can be adjusted to weight in favour of insertions, deletions or substitutions, or to favour contiguous deletions, as in the first *waiter–waitress* conversion, over non-contiguous ones. In fact the sequence-comparison algorithm developed by Levenshtein, which compares any sequences of symbols — characters, words, digits, whatever — has a huge number of applications, ranging from file comparison in computers, to speech recognition (sound waves can be represented as sequences of digits), comparison of genetic sequences such as DNA, image processing … in fact anything that can be digitised can be compared using Levenshtein distance.

A drawback with this simplistic string-edit distance is that it does not take other factors into account. For example, consider the four sentences in (8).

- (8) a. Select ‘Symbol’ in the Insert menu.
- b. Select ‘Symbol’ in the Insert menu to enter a character from the symbol set.
- c. Select ‘Paste’ in the Edit menu.
- d. Select ‘Paste’ in the Edit menu to enter some text from the clip board.

Given (8a) as input, most character-based similarity metrics would choose (8c) as the best match, since it differs in only two words, whereas (8b) has eight additional words. But intuitively (8b) is a better match since it entirely includes the text of (8a). Furthermore (8b) and (8d) are more similar than (8a) and (8c): the latter pair may have fewer words different (2 vs. 6), but the former pair have more words in common (8 vs. 4), so the distance measure should count not only differences but also similarities.

4.2 More sophisticated matching

The similarity measure in the TM system may be based on individual characters or whole words, or may take both into consideration. One could certainly envisage more sophisticated methods, incorporating linguistic “knowledge” of inflection paradigms, synonyms and even grammatical alternations, though it is unclear whether any existing systems go this far. To exemplify, consider (9a). The example

(9b) differs only in a few characters, and would be picked up by any currently available TM matcher. (9c) is superficially quite dissimilar, but is made up of words which are related to the words in (9a) either as grammatical alternatives or near synonyms. (9d) is very similar in meaning to (9a), but quite different in structure. Arguably, any of (9b–d) should be picked up by a sophisticated TM matcher, but it is unlikely that any commercial TM system would have this capability.

- (9)
 - a. When the paper tray is empty, remove it and refill it with paper of the appropriate size.
 - b. When the tray is empty, remove it and fill it with the appropriate paper.
 - c. When the bulb remains unlit, remove it and replace it with a new bulb
 - d. You have to remove the paper tray in order to refill it when it is empty.

The reason for this is quite important. The matcher uses a quite generic algorithm, as mentioned above. If we wanted it to make the kind of more sophisticated *linguistically*-motivated distinctions involved in the examples in (9), the matcher would have to have some language-specific “knowledge”, for example about which words were more or less important for the match, parts of speech and meanings of individual words and, in the case of (9d) knowledge about equivalent meanings of different syntactic structures. In short, the matcher would have to know what language the text was written in, and would have to be different for different languages. It is doubtful whether the gain in accuracy (see below) would merit the extra effort required by the developers. As it stands, TM systems remain largely independent of the source language and of course wholly independent of the target language. We will discuss below how to evaluate the matching algorithm in a TM system.

4.3 Segment (fragment) matching

Nearly all TM systems work exclusively at the level of sentence matching. But consider the case where an input such as (10) results in matches like those in (11).

- (10) Select ‘Symbol’ in the Insert menu to enter a character from the symbol set.
- (11)
 - a. Select ‘Paste’ in the Edit menu.
 - b. To enter a symbol character, choose the Insert menu and select ‘Symbol’.

Neither match covers the input sentence sufficiently, but between them they contain the answer. It would clearly be of great help to the translator if TM systems could

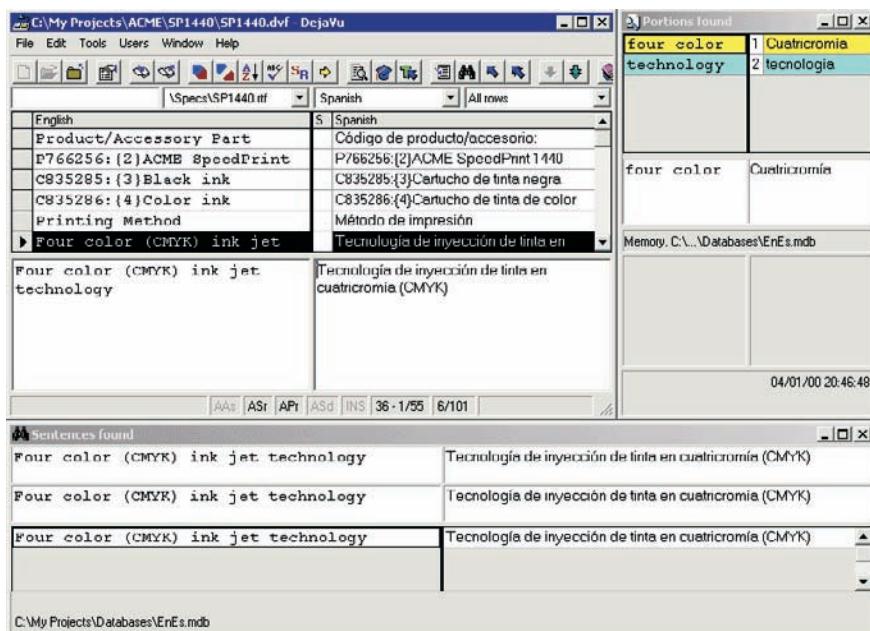


Figure 5. “Portion matching” in *Déjà Vu*.

present partial matches and allow the user to cut and paste fragments from each of the matches. This has been called “Shallow TM” by some commentators, and is being worked on by most of the companies offering TM products, and, in a simplified form, is currently offered by at least one of them, *Déjà Vu*. Figure 5 shows its “Portion matching” window, in which any words or groups of words found in the lexicon, the terminology database or the memory database itself are displayed.

5. Evaluation

TM systems are undoubtedly the most popular translator’s aid to have emerged from research and development in CAT and MT, and there is an irony here because actually they are the least sophisticated of all the ideas that have been tried. Popular though they may be, how good are they? Evaluation is a major issue in any software-related enterprise, and, as Chapter 13 shows, evaluating CAT and MT systems is a major preoccupation. In this section we will look briefly at some of the factors involved in evaluating TM systems.

5.1 Evaluating user friendliness

Like most software systems, user friendliness is a major issue. This includes issues like how easy it is to install the software and to get it running, how seamlessly it can be integrated into the word-processing package already in use and, above all, whether it does what the user expects it to do.

Other issues include the quality of the **documentation**: how easy is it to find out how to do something and/or what the particular function of any part of the software is. Notice that these two perspectives on documentation are quite different (and the distinction applies to any kind of software, not just translation-related products). Too many user manuals list the functions of the software, going through them in the order that they appear in the drop-down menus, sometimes in a naively simplistic way (e.g. *The 'Open File' command is used to open a file*), whereas it may be more useful for the user to have a task-oriented view of the software: *How do I choose to show fewer fuzzy matches?*

5.2 Evaluating productivity gains

Above all, the potential purchaser wants to know whether and how much using a TM will lead to a gain in productivity. One of the major reasons, as Michael Benis points out, that some users are disappointed by the relatively small productivity increases achieved is that they do not receive enough work in electronic format or of a sufficiently similar nature for a TM system to make a significant impact on their earnings: these are prerequisites to getting the best out of TM. In certain cases, however, it is also because they have been misinformed about the scale of productivity increases that are possible: while on occasion a TM product might result in a 60% productivity increase, it would be unreasonable to expect this kind of gain every time you use it, and 30% may be a more reasonable average expectation (which of course means that sometimes you will get much less than 30%). It is also likely that productivity will increase the more the software is used, both because of the familiarity of the user and the increased size of the underlying data, but you can expect this gain to tail off at some point.

Another highly sensitive issue is how much customers should be charged for a translation done with a TM. All translators have experience of clients who understand so little of what is involved in translation that they can have ridiculous expectations both in terms of time and effort needed, and the likely cost. But equally, some customers are well aware of the technology available, and know how TM can be used for highly repetitive texts, or to update translations of previously translated texts. In this situation, translators should make their customers aware of how the TM is used, and the extent to which it actually saves effort, as opposed to

transferring effort. For example, while cutting and pasting 100% matches is relatively easy, it does not require no effort at all, as the translator must still check that the translation offered is appropriate. Similarly, adapting partial matches to the new input is not the same as translating from scratch, but the amount of work may or may not be directly proportional to the percentage of fuzzy match on the source text. Remember too that you will need to invest considerable effort in setting up your TM system and learning how to get the best out of it, a cost which you will not be able to pass on to your customers.

5.3 Evaluating the matching algorithm

At a more fine-grained level, some researchers (and users) are interested in evaluating exactly what the potential for time-saving a TM system represents, by trying to quantify the relationship between the fuzzy match scores and the amount of typing the translator is spared.

This question has two aspects. First, do the fuzzy match scores, which are based on matching the *source*-language segments, accurately correspond to similarity in the target-language segments which are proposed? And second, at what level of similarity is it still quicker to edit the proposed matches rather than simply translate from scratch?

Both questions are actually difficult to resolve definitively. Some informal research by this author and my students has suggested that at the top end, the fuzzy match scores probably do give, for practical purposes, as good a measure of similarity with matches in the database as could be desired. Various different similarity measures can be tried but they usually give the same answer. Occasionally the second- or third-rated match may prove to be more useful to the translator, and the lower limits of usefulness probably depend on how similar, in linguistic terms, are the source and target languages.

To answer the second question, we set up some experiments and counted the keystrokes needed to edit the proposed matches into the target text. Even counting keystrokes is not straightforward since, with modern word-processors, there are always alternative methods of achieving the same result using different combinations of keyboard, mouse moves and hot keys. Nevertheless we found that the threshold of usefulness is not particularly low, and as the fuzzy match score drops below the 75% mark, and the number of matches offered exceeds a half a dozen, the usefulness diminishes rapidly.

6. Example-based MT

TM systems are often associated with — confused with, it might even be said — an approach to MT called “Example-based MT” (EBMT). The present author believes that there are important differences between TMs and EBMT, but the two approaches do share some basically similar underlying ideas which are worth exploring briefly here.

In EBMT, like in TMs, there is an aligned parallel corpus of previous translations, and from this corpus are selected appropriate matches to the given input sentence. In a TM, however, it is up to the user, the translator, to decide what to do with the retrieved matches. In EBMT, we try to automate the process of selecting the best matches or fragments from the best matches, and then to “recombine” the corresponding target-language fragments to form the translation. Because this has to be done automatically by the system, any linguistic knowledge or translator’s expertise that needs to be brought to bear on the decision has to be somehow incorporated into the system. We can illustrate this with a kind of exercise for the reader.⁴ Consider the two translation pairs given in (12) and (13).

- (12) The monkey ate a peach. \Leftrightarrow *saru wa momo o tabeta.*
(13) The man ate a peach. \Leftrightarrow *hito wa momo o tabeta.*

Without any knowledge of the language concerned, it is reasonable to assume that the difference between the English sentences on the left corresponds to the difference between the Japanese translations: in (13) we substitute *man* for *monkey* and *hito* for *saru*, so it is not unreasonable to assume (14):

- (14) monkey \Leftrightarrow *saru*; man \Leftrightarrow *hito*

Actually, we can make a further assumption, which is that the “remainder” of the two sentences also correspond (15).

- (15) The ... ate a peach. \Leftrightarrow ... *wa momo o tabeta.*

Now if we look at some further evidence (16), what can we conclude?

- (16) The dog ate a rabbit. \Leftrightarrow *inu wa usagi o tabeta.*

In (16) we have two new word pairs, *dog* and *rabbit* in English, *inu* and *usagi* in Japanese. We have no direct evidence but a kind of circumstantial evidence, based on our knowledge that languages tend to be systematic in these kinds of things, that *dog* corresponds to *inu*, and *rabbit* to *usagi*. Furthermore, if that is right, we can also say now that *peach* is *momo*, giving us (17) as a residue.

- (17) The ... ate a.... \Leftrightarrow ... *wa ... o tabeta.*

So far, all our conclusions have been based on corresponding substitutions in similar pairs of matches, and we could expect to be able to use this knowledge to “recombine” the elements to produce, correctly, the translation pair in (18) and other combinations.

- (18) The dog ate a peach. \Leftrightarrow *inu wa momo o tabeta*.

What now if we wanted to translate (19a)? We know the words for *man* and *dog*, and the template in (17). It might be reasonable, on the evidence alone, to imagine that the little *wa* and *o* words correspond to the little *the* and *a* words, and that *tabeta* is *ate*: if we put it all together we get (19b). Actually, we do not have any direct evidence for that, just an instinct perhaps. Depending on our experience of other languages, we might just as easily have the instinct that *wa* indicates the subject and *o* the object, giving us (19c), and that the translation of *the* and *a* is hidden somehow in the combination of words in (17). In fact, all that we know for sure is captured in (17): strictly speaking, we do not really know how the translation of the words *the*, *a* and *ate* is distributed amongst the Japanese words in (17).⁵

- (19) a. A man ate the dog.
 b. *hito o inu wa tabeta*.
 c. *hito wa inu o tabeta*.

The point of this elaborate example is to show some of the pitfalls in EBMT and how it differs from TM. While the two tools share the matching function and the database of previous examples, it is the use that is made of the matched examples that is the driving force: in a TM system it is a human that makes these delicate decisions which require so much more than simply sticking bits together: knowledge of how the source and target strings relate to each other, and how to render grammatical sentences in the target language are required. Nevertheless these are interesting and challenging problems for computational linguists, and research into EBMT as a way of realising the dream of MT continues.

Further reading

The history of TMs is told as part of the story of the development of the translator’s workbench by Hutchins (1998). The *ITI Bulletin* has published two extensive articles on TMs (Holloway, 1996; Benis, 1999) and will probably continue to do so periodically. Bowker (2002) includes an extensive chapter on TMs.

For information on LISA, try its web site www.lisa.org, where more information about TMX can also be found (www.lisa.org/tmx).

There is an extensive literature on alignment, much of it focusing on the mathematical, statistical and computational aspects; perhaps the most accessible overview is Manning and Schütze (1999), pp. 466–486.

Evaluations of TM systems are found in Holloway (1996) and Benis (1999), already mentioned. TM evaluation techniques also form the basis of a case study by the EU's Expert Advisory Group on Language Engineering Standards (EAGLES, 1996).

Somers (1999) provides an extensive review of EBMT.

Notes

1. The term “translation memory” is used for both the generic software — more properly “translation memory system” — and the database of previous translations, the “memory” itself. To avoid confusion, we will use the term “database” for the latter (preferring it over the term “memory” which has a rather specific meaning for computer scientists).
2. Taken from the *Trados WinAlign* tool. The aligned text is partially obscured for proprietary reasons.
3. A “distance” measure is simply the inverse of a “similarity” measure.
4. Apologies to readers who know Japanese and who will not be able to do this exercise with the necessary degree of ignorance!
5. Of the Japanese translations offered in (19), (19c) is the correct one: *wa* and *o* are (roughly speaking) case markers.

References

- Arthern, P. J. (1981) “Aids Unlimited: The Scope for Machine Aids in a Large Organization”, *Aslib Proceedings* 33, 309–319.
- Benis, Michael (1999) “Translation Memory from O to R”, *ITI Bulletin*, April 1999, 4–19; also available (updated) on the web at www.star-group.net/press/tm-review01.htm.
- Bowker, Lynn (2002) *Computer-aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.
- EAGLES (1996) “Benchmarking Translation Memories”, in *Evaluation of Natural Language Processing Systems, Final Report* (EAG-EWG-PR.2), Commission of the EU, Luxembourg, www.issco.unige.ch/projects/ewg96/node157.html.
- Holloway, Trevor (1996) “Computer-Assisted TransMogrification?”, *ITI Bulletin*, August 1996, 16–25.
- Hutchins, John (1998) “The Origins of the Translator’s Workstation”, *Machine Translation* 13, 287–307.
- Kay, Martin (1980) “The Proper Place of Men and Machines in Language Translation”,

- Research Report CSL-80-11, Xerox PARC, Palo Alto, Calif. Reprinted in *Machine Translation* 12 (1997), 3–23.
- Manning, Christopher D. and Hinrich Schütze (1999) *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts.
- Melby, Alan K. (1995) *The Possibility of Language: A Discussion of the Nature of Language*. John Benjamins, Amsterdam.
- Somers, Harold (1999) “Review Article: Example-based Machine Translation”, *Machine Translation* 14, 113–157.

CHAPTER 4

Terminology tools for translators

Lynne Bowker
University of Ottawa, Canada

1. Introduction

Terminology is the discipline concerned with the collection, processing, description and presentation of **terms**, which are lexical items belonging to specialized subject fields. Identifying equivalents for specialized terms is a major part of any translation project. Subject fields such as engineering, physics, medicine, law, etc., all have significant amounts of field-specific terminology. In addition, many clients will have preferred in-house terminology. It can be a time-consuming and labour-intensive task to research the specific terms needed to complete any given translation, and translators do not want to have to repeat all this work each time they begin a new translation. There are a number of different types of computer tools that can help with various aspects of the translator's terminology-related tasks, including the storage, retrieval and updating of term records. By using terminology tools, translators can help to ensure greater consistency in the use of terminology, which not only makes documents easier to read and understand, but also prevents miscommunications. Effective terminology management can help to cut costs, improve linguistic quality, and reduce turn-around times for translation, which is very important in this age of intense time-to-market pressures.

The aim of this chapter is to present some different types of terminology tools that can be useful for translators. Section 2 opens with a brief history of the use of computer tools in terminology. In Section 3, we focus on what type of information can be found in term records, noting that human translators and machine users have different needs. Features of contemporary terminology-management tools are examined in Section 4, and some of the benefits of working with such tools are explored in Section 5. Finally, Section 6 introduces another type of computer-aided terminology tool — a term-extraction tool — which has been under development for some time, but which has only recently become widely commercially available.

2. A brief history of terminology tools

Terminology tools have been in existence for some time. Dating back to the 1960s, term banks were among the first linguistic applications of computers. Term banks are basically large-scale collections of electronic term records, which are entries that contain information about terms and the concepts they represent (e.g., definitions, contexts, synonyms, foreign language equivalents, grammatical information). Early term banks were originally developed by large corporations or institutions to serve as resources for in-house translators. Translators still constitute the primary user group of such resources, though the contents of many term banks are now made available to a wider audience, including freelance translators and translation agencies. Some term banks can be accessed freely on the World Wide Web, while others are available via subscription and may be distributed on CD-ROM. Some well-known term banks include *Eurodicautom*, *Termium*, *Normaterm*, and the *Grand dictionnaire terminologique* (formerly the *Banque de terminologie du Québec*).¹ These were among the first term banks to be developed, and they are still in existence today, although they have evolved in terms of their contents and appearance.

Because term banks endeavour to serve a wide range of translators, they are almost always multilingual and they typically cover a broad array of specialized subject fields. While the aim is generally to produce a detailed record for each term (i.e., containing both linguistic and extra-linguistic information), some records are more detailed than others. Term banks are a very dynamic resource and they are updated frequently. Most institutions that maintain term banks also have a team of terminologists who conduct terminological research and compile the term records. Users, such as translators, may be invited to submit data for possible inclusion in the term bank, but this data is always vetted by the term bank's quality control officers.

There is no doubt that term banks constitute valuable translation resources; however, since specialized subject fields and the language used to describe these fields are constantly expanding and evolving, it is not possible for any term bank to provide exhaustive up-to-date coverage. Moreover, clients may have terminological preferences that are not reflected in the term banks maintained by other institutions. Therefore, most translators find that it is necessary to compile their own terminological records in order to ensure that the appropriate subject fields and client preferences are adequately represented. There are a number of different options for managing personal terminology collections, ranging from non-technological techniques to sophisticated computer programs. For example, translators can create and manage glossaries using index cards, word processors, spreadsheets, databases, or specially designed terminology management systems (TMSs). This chapter will focus on the last of these options.²

When desktop computers first became available in the 1980s, personal TMSs were among the first computer-aided translation (CAT) tools to be made commercially available to translators. Translators were able to use these tools to create and maintain personal **termbases**, in which they could record the results of their own terminological research. Although they were very welcome at the time, these early TMSs had a number of limitations. For instance, they were designed to run on a single computer and could not easily be shared with colleagues or clients. Moreover, they typically allowed only simple management of bilingual terminology and imposed considerable restrictions on the type and number of data fields as well as on the maximum amount of data that could be stored in these fields. Recently, however, this type of software has become more powerful and flexible, as we will see in Section 4.

One of the newest computer-aided terminology tools to arrive on the scene is the **term-extraction** tool. Essentially, this type of tool attempts to search through an electronic corpus (see Chapter 7) and extract a list of candidate terms that a translator may wish to include in a termbase. This process will be described in more detail in Section 6. Term-extraction tools have been the object of research and development for some time now, but it is only relatively recently that they have become commercially available on a wide scale. It is becoming increasingly common to see such tools included in translator's workstations (see Chapter 2), alongside tools such as terminology management systems, translation memories, and concordancers. Related tools, such as those that will attempt to identify automatically other types of terminological information in corpora, such as collocations, definitions, synonyms and conceptual relations, are also under active development.³

3. What goes into a term record?

Both term banks and termbases are made up of data records called **term records**. Term records treat a single concept and may contain a variety of linguistic and extra-linguistic information associated with that concept in one or more languages. There are no hard-and-fast rules about what kind of information should be included on a term record — translators will have to decide this for themselves based on the availability of data and on the requirements of the project at hand. Nevertheless, types of information that may be found on term records could include: an indication of the subject field, equivalents in one or more languages, grammatical information (e.g., part of speech or gender), synonyms, definitions, contexts, usage notes (e.g., rare, archaic, British), and any other comments or information the translator thinks might be helpful in order to use the term in question correctly.

One common difference between term banks and termbases is that the former

strive to complete detailed records in order to meet the needs of a wide range of users. In contrast, the records in termbases are generally for the personal use of the translator who creates them; therefore, these records are frequently less detailed and may contain only those pieces of information that translators find useful or relevant to their needs.

For example, although TMSs do allow users to enter detailed information, it is becoming increasingly common to see termbases used in the localisation industry (see Chapter 5) that contain only the source and target term, and perhaps a comment if the source term has multiple possible translations depending on the context. Some of the reasons for this type of stripped-down term record format include the following (see O'Brien, 1998: 118). Firstly, the required turn-around time in the localisation industry is often so short that it does not allow for the preparation of detailed glossaries. In addition, the terminology that is used — even by the same client — can change rapidly, warranting new glossaries each time the client has a product localised. Finally, the translator, who also has to produce very fast turn-around times, is interested only in the client-approved translated term and the context in which a term can occur if there is more than one translation for the same term. The fact that technology makes it easy to compile and transfer information quickly has contributed to this trend of treating termbases as disposable items, rather than as long-standing records.

3.1 Terminology resources for machine users

Term banks and termbases are typically intended for use by human translators; however, there is another type of user that may also need terminological information — a computer. For example, machine translation systems that operate in specialized subject fields need both general language dictionaries and specialized terminology resources. There is a difference, however, in the type of information that is needed by human and machine users.

As discussed above, a human translator may create a term record that contains only a few pieces of information relating to the term in question, such as its foreign language equivalent and a definition or context. Additional information, such as grammatical information, may not be required if the translator is already familiar with the grammar of the languages in question.

In contrast, the type of information needed by a machine is very different. Machines are not intelligent and will not be able to understand definitions or contextual examples, nor will they have an innate knowledge of grammatical systems or of real-world situations. Detailed grammatical information, such as part of speech, gender, and number must be explicitly recorded in a highly structured

way in machine-readable terminology resources. Morphological data, particularly for irregular plural formations or verb conjugations, will also be required by machines. Other types of specialized information, including subcategorization features, semantic features, selectional restrictions, valency information, and case frames may be needed in order to help a machine translation system use terminology correctly.

Other types of computer tools, such as information retrieval systems, also use machine-readable terminology resources, such as thesauri or controlled vocabularies (see, for example, Strehlow, 2001).

4. A new generation of Terminology-Management tools

As mentioned previously, a TMS is a computer program that allows a user, such as a translator, to create a personal termbase. In the following sections, we will explore the principal features of contemporary TMSs, focussing on improvements that have been made with regard to storage, retrieval and integration with other CAT tools.

4.1 Storage

The most fundamental function of a TMS is that it acts as a repository for consolidating and storing terminological information for use in future translation projects. In the past, many TMSs stored information in structured text files, mapping source to target terminology using a unidirectional one-to-one correspondence. This caused difficulties, for example, if an English–French termbase needed to be used for a French–English translation. Contemporary TMSs tend to store information in a more concept-based way, which permits mapping in multiple language directions.

There has also been an increased flexibility in the type and amount of information that can be stored on a term record. Formerly, users were required to choose from a pre-defined set of fields (e.g., subject field, definition, context, source), which had to be filled in on each term record. In addition, the number of fields was often fixed, as was the number of characters that could be stored in each field. For instance, if a TMS allowed for only one context, then the translator was forced to record only one context, even though it may have been useful to provide several. An example of a typical conventional record template is provided in Figure 1.

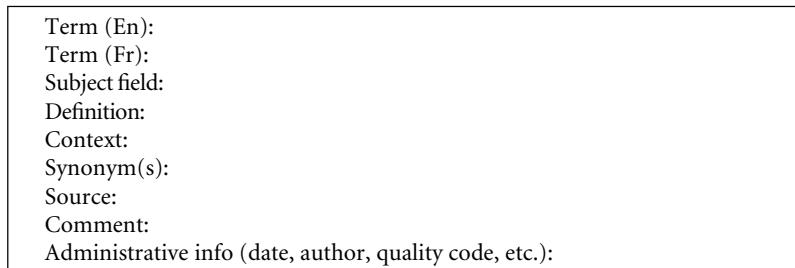


Figure 1. Conventional TMSs came with a fixed set of pre-defined fields.

In contrast, as illustrated in Figure 2, most contemporary TMSs have adopted a **free entry structure**, which allows translators to define their own fields of information, including repeatable fields (e.g., for multiple contexts or clients). Some TMSs even permit the inclusion of graphics. Not only can translators choose their own information fields, they can also arrange and format them, choosing different fonts, colours or layouts for easy identification of important information. This means that the software can be adapted to suit a specific translator's needs and the termbase can evolve as future requirements change. The amount of information that can be

The screenshot shows the 'TermBase Editor' window of the 'MultiTrans Demo' application. The menu bar includes File, Edit, Process, View, TermBase, Window, and Help. The main window has tabs for Details and Search, currently showing the Details tab. The left sidebar lists various actions: copy, copy and paste, cut, cut and paste, edit, find, global search and replace, paste, replace, and search. The central area displays two sections: 'Expression' and 'Translations'. The 'Expression' section contains a 'global search and replace' dialog with fields for Context1, Context2, Source1, and Source2, along with a note about performing a global search and replace. It also shows a table of creation details: Created date 17-09-2001, Created by Demo User, Modified date 17-09-2001, Modified by Demo User, and Tag Accepted(♦). The 'Translations' section shows two entries: 'recherche et remplacement automatique (French)' and 'recherche et substitution automatique (French)'. Each entry has a table with fields: Frequency (1), Source (Magazine informatique XYZ), Client (Company A/B), and a table of creation details. There are 'View | Modify' links next to each entry.

Figure 2. Flexible TMSs, such as *TermBase* from MultiCorpora, allow translators to create and organize their own information fields.

stored in any given field or on any given record has also increased dramatically. Different termbases can be created and maintained for different subject fields or different clients, and some systems allow multiple termbases to be merged if desired.

4.2 Retrieval

Once the terminology has been stored, translators need to be able to retrieve this information. A range of search and retrieval mechanisms is available. The simplest search technique consists of a simple look-up to retrieve an exact match. Some TMSs permit the use of wildcards for truncated searches. A wildcard is a character such as an asterisk (*) that can be used to represent any other character or string of characters. For instance, a **wildcard search** using the search string *translat** could be used to retrieve the term record for *translator* or the term record for *translation*, etc. More sophisticated TMSs also employ **fuzzy matching** techniques. A fuzzy match will retrieve those term records that are similar to the requested search pattern, but which do not match it exactly.

Fuzzy matching allows translators to retrieve records for morphological variants (e.g., different forms of verbs, words with suffixes or prefixes), for spelling variants (or even spelling errors), and for multiword terms, even if the translators do not know the precise order of the elements in the multiword term. Some examples of term records that could be retrieved using fuzzy matching techniques are illustrated in Figure 3.

<i>Search pattern entered by user:</i>	<i>Term record retrieved using fuzzy matching:</i>
advertising organisation centre for preventing and controlling diseases	advertisement organization Center for Disease Control and Prevention

Figure 3. Term records retrieved using fuzzy matching.

In cases where wildcard searching or fuzzy matching is used, it is possible that more than one record will be retrieved as a potential match. When this happens, translators are presented with a **hit list** of all the records in the termbase that may be of interest and they can select the record(s) they wish to view. Figure 4 shows some sample hit lists.

<i>Hit list containing records that match the wildcard search pattern ‘*nut’</i>	<i>Hit list containing records that match the fuzzy search pattern ‘post-office box number’</i>
coconut hazelnut peanut walnut	PostOffice post office box P. O. box number postbox

Figure 4. Sample hit lists retrieved for different search patterns.

4.2.1 Automatic terminology lookup and pre-translation

Another type of specialized retrieval feature offered by some TMSs, particularly those that operate as part of an integrated package with a word processor and translation memory (see Chapter 3) is known as **automatic terminology lookup**. This feature is essentially a type of automatic dictionary lookup. As the translator moves through the text, the terminology recognition component automatically compares lexical items in the source text against the contents of the termbase. As shown in Figure 5, if a term is recognized as being in the termbase, the translator's attention is drawn to the fact that a record exists for this term, and the translator can then view the term record and can copy and paste the term from the record directly into the target text.

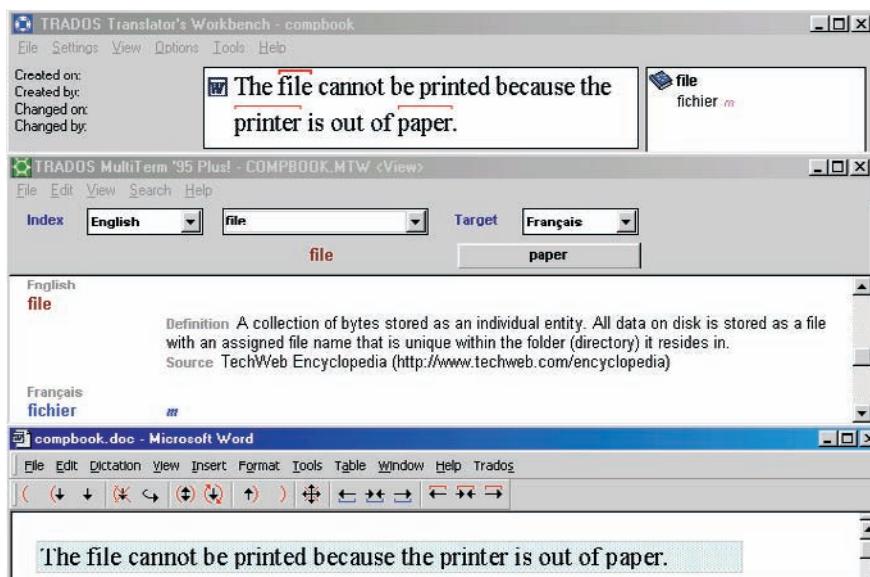


Figure 5. Automatic terminology lookup in *Trados*.

Some TMSs also permit a more automated extension of this feature where a translator can ask the system to do a sort of pre-translation or batch processing of the text. In the case of pre-translation, the TMS will identify those terms for which an entry exists in the termbase, and it will then automatically insert the corresponding equivalents into the target text. The output of this pre-translation phase is a sort of hybrid text, as shown in the bottom right-hand corner of Figure 6. In a post-editing phase, the translator must verify the correctness of the proposed terms and translate the remainder of the text for which no equivalents were found in the termbase.

4.3 Additional features

Most TMSs can operate as standalone applications; however, as mentioned previously, many contemporary systems can also be integrated with other products, particularly translation memory systems and word processors. TMSs may also include other types of utilities, such as features that allow users to create and manage concept systems or thesauri, to merge multiple termbases, to import from and export to other formats, or to print out the contents of a termbase in a user-specified glossary format. The precise features available will, of course, depend on the specific product in question.

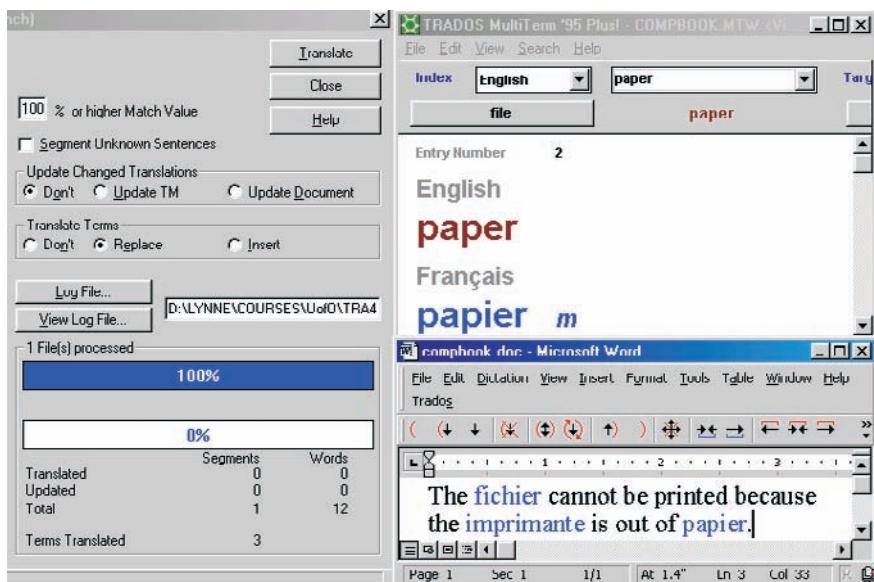


Figure 6. A hybrid text produced as a result of pre-translation in *Trados*.

5. Benefits of working with a TMS

The idea behind maintaining a glossary of any kind is that it encourages terminological consistency and prevents translators from having to repeat their research each time they start a new translation project. As already mentioned, it is not necessary to use specialized software to maintain a glossary — many translators have long been using card indexes or word processors to create terminology records. However, a TMS does offer a number of advantages over these conventional approaches.

5.1 Quality

Although any type of glossary can help to improve consistency throughout a translation project, the automatic terminology lookup feature of some TMSs takes this one step further. After all, there is not much point in going to the trouble of ensuring that terminology is agreed beforehand and stored in a termbase if translators choose not to consult this termbase (see Clark, 1994: 306). With automatic terminology lookup, the choice is taken out of the translator's hands because terms in the source text are automatically checked against the termbase.

5.2 Speed and flexibility

The principal advantages of using a TMS rather than a card index have largely been outlined above in Sections 4.1 and 4.2: TMSs permit more flexible storage and retrieval. In addition, it is easier to update electronic information, and faster to search through electronic files. Even though a word processor allows information to be stored in electronic form, it is not an adequate tool for managing terminology in an efficient way, and its search facilities slow down considerably as the termbase grows in size.

Another way that a TMS can potentially speed up a translator's work is by allowing terms to be pasted directly into the target text, thus avoiding the need to retype the term. Of course, some editing may be required (e.g., to conjugate a verb), and this has raised an interesting question with regard to which form of a term should be recorded on a term record. Traditionally, terminologists have been encouraged to record the “canonical form” of a term (e.g., the nominative singular form of a noun, the infinitive form of a verb, the masculine singular form of an adjective) on the term record (for example, see Rondeau, 1984: 84; Dubuc, 1985: 80). However, in order to reduce the amount of time spent editing terms that

have been inserted directly from TMSs, some translators (see Kenny, 1999: 71) are now choosing to record the most common form of a term, or indeed several forms of a term, as shown in Figure 7. This way, the correct form can be inserted simply by clicking on it, and there will be no need to edit the term in the target text.

En:	print printed printing
Fr:	imprimer imprimez imprimé imprimée imprimées imprimés

Figure 7. Multiple forms of the term can be recorded on a term record to facilitate automatic insertion of the required form directly into the target text.

5.3 Shareability of information

TMSs can be used as standalone tools, but more and more, they are being networked so that several translators can access and contribute to a single termbase. This option can help to ensure consistency on projects where several translators may be working on different parts of a long document. In such cases, it may be desirable to give different users different types of privileges on the network system. For instance, all users should be able to consult the information stored in the termbase, but only some users, such as those translators responsible for quality control, can add new records.

Another way of sharing terminological information is by exchanging data with clients or with other translators. Unfortunately, not everyone uses the same TMS, and different applications store information in different formats. In order to exchange information the file formats must either be compatible or convertible. Some TMSs will allow data to be exported directly to various word processor or desktop publishing system formats. Some TMSs also allow data to be imported and exported according to international standards, such as the Machine-Readable Terminology Interchange Format (MARTIF – ISO 12200; see Melby et al., 2001). A new standard, known as Term Base eXchange (TBX), has recently been developed by the Open Standards for Container/Content Allowing Reuse (OSCAR) special interest group of the Localisation Industry Standards Association (LISA).⁴

6. Term-extraction tools

Another type of computer-aided terminology tool that is now gaining popularity with translators is the **term-extraction tool**, sometimes referred to as a term-identification or term-recognition tool. Basically, this type of tool can help translators to get a head start on building up a termbase by searching through electronic corpora (see Chapter 7) and extracting lists of potential terms that translators may wish to include in their termbase.

Term-extraction tools can be either monolingual or bilingual. A monolingual tool attempts to analyze a text or corpus in order to identify candidate terms, while a bilingual tool analyzes existing source texts along with their translations in an attempt to identify potential terms and their equivalents. Although the initial extraction attempt is performed by a computer program, the resulting list of candidates must be verified by a human terminologist or translator, and for this reason, the process is best described as being computer-aided or semi-automatic rather than fully automatic.

There are two main approaches to term extraction: linguistic and statistical. For clarity, these approaches will be explained separately; however, aspects of both approaches can be combined in a hybrid term-extraction tool.

6.1 Linguistic approach to term extraction

Term-extraction tools that use a linguistic approach typically attempt to identify word combinations that match particular part-of-speech patterns. For example, in English, many terms consist of adjective+noun or noun+noun combinations. In order to implement such an approach, each word in the corpus must first be associated with its appropriate part of speech, which can be done with the help of a piece of software known as a **tagger**. Once the corpus has been tagged, the term-extraction tool simply identifies all the occurrences that match the specified part-of-speech patterns. For instance, a tool that has been programmed to identify adjective+noun and noun+noun combinations as potential terms would identify all lexical combinations matching those patterns from a given corpus, as illustrated in (1), where candidate terms are italicized.

- (1) For *assisted reproduction*, *egg collection* is usually performed with the help of ultrasound. To accomplish this, a needle is inserted through the *vaginal wall* into the ovaries using ultrasound to locate each follicle.

Unfortunately, not all texts can be processed this neatly. If the corpus is modified slightly, as illustrated in (2), problems such as **noise** and **silence** may occur.

-
- (2) For *in vitro fertilization*, *egg collection* is usually performed with the help of ultrasound. To accomplish this *delicate task*, a *small needle* is inserted through the *vaginal wall* into the ovaries using ultrasound to locate each follicle.

First, not all of the combinations that follow the specified patterns will qualify as terms. Of the noun+noun and adjective+noun candidates that were identified in (2), some qualify as terms (*egg collection*, *vaginal wall*), whereas others do not (*delicate task*, *small needle*). The latter set constitutes “noise” and would need to be eliminated from the list of candidates by a human.

Another potential problem is that some legitimate terms may be formed according to patterns that have not been pre-programmed into the tool. This can result in “silence”—a situation where relevant information is not retrieved. For example, the partial term *vitro fertilization* has been identified as a candidate because it matches the pattern noun+noun, but the actual term is *in vitro fertilization*, which is formed using the pattern preposition+noun+noun—a pattern that is not very common and therefore may not be recognized by term-extraction tools.

An additional drawback to the linguistic approach is that it is heavily language dependent. Term-formation patterns differ from language to language. For instance term-formation patterns that are typical in English (e.g., noun+noun, adjective+noun) are not the same as term-formation patterns that are common in French (e.g., noun+preposition+noun, noun+adjective). Consequently, term-extraction tools that use a linguistic approach are generally designed to work in a single language (or closely related languages) and cannot easily be extended to work with other languages.

6.2 Statistical approach to term extraction

Term-extraction tools that use a statistical approach basically look for repeated sequences of lexical items. The **frequency threshold**, which refers to the number of times that a sequence of words must be repeated, can often be specified by the user. For instance, as shown in (3), if the minimum frequency threshold is set at 2, then a given sequence of lexical items must appear in the corpus at least twice in order to be recognized as a candidate term by the term-extraction tool.

- (3) *Injectable fertility medications* used to boost egg production prior to *in vitro fertilization* may include *follicle stimulating hormone*, *luteinizing hormone* and *human chorionic gonadotropin*. *Follicle stimulating hormone* is given as a subcutaneous injection while *human chorionic gonadotropin* and *luteinizing hormone* are administered as intramuscular injections. Risks associated with *injectable fertility medications* might include swelling or bruising.

Based on a minimum frequency threshold of 2, the corpus in (3) yielded four potential terms: *injectable fertility medications*, *follicle stimulating hormone*, *luteinizing hormone* and *human chorionic gonadotropin*. Unfortunately, this simple strategy often leads to problems because language is full of repetition, but not all repeated sequences of lexical items qualify as terms. For example, consider the slightly modified version of (3) that appears in (4).

- (4) *Injectable fertility medications* used to boost egg production prior to in vitro fertilization *may include follicle stimulating hormone, luteinizing hormone and human chorionic gonadotropin. Follicle stimulating hormone is given as a subcutaneous injection while human chorionic gonadotropin and luteinizing hormone are given as intramuscular injections.* Risks associated with *injectable fertility medications may include* swelling or bruising.

Working solely on the basis of identifying repeated sequences of lexical items, the term-extraction software has identified two additional candidates: *may include* and *given as*. Unfortunately, these candidates constitute noise rather than terms, and as such they would need to be eliminated from the list of potential terms by a human.

Another strategy for reducing the number of unlikely terms that may otherwise be identified as candidates is to use a **stop list**, that is, a list of items that the computer can be instructed to ignore. For instance, a stop list could be used to instruct the term-extraction tool to ignore any sequence of lexical items that begins or ends with a function word (e.g., articles, prepositions, conjunctions), since these are not likely to constitute terms. There are, however, exceptions as in the case of the term *in vitro fertilization*, which begins with a preposition.

Another drawback to the statistical approach is that not all of the terms that appear in a given text will be repeated, which may lead to silence. For instance, in (4), the term *in vitro fertilization* was not identified as a candidate because it appeared in the corpus only once and the minimum frequency threshold was set to 2.

Nevertheless, the statistical approach does have one clear strength: because it works by identifying repeated patterns, it is not language dependent. This means that a term-extraction tool employing this approach can, in principle, be used to process texts in multiple languages.

7. Conclusion

Identifying and using the correct terminology is an extremely important part of any translation project. In this chapter, we have tried to show how computer-aided terminology tools, such as TMSs and term-extraction tools, can be used to help translators carry out their terminological research more efficiently and apply their

findings more consistently, which will in turn result in faster turn-around times and increased quality.

Further reading

The discipline of terminology in general is covered by works such as Sager (1990), Wright and Budin (1997) and Cabré (1999). Historical overviews of the development of term banks can be found in Rondeau (1984) and Sager (1990), while an overview of TMSs can be found in Schmitz (1996). More information about terminology and MT can be found in Koch (1995) and Vasconcellos (2001). Lauriston (1997) and Schmitz (2001) outline criteria for evaluating contemporary TMSs. Term-extraction tools are discussed by Kageura and Umino (1996), Lauriston (1997), L'Homme (1999), Jacquemin (2001), Ahmad and Rogers (2001) and Cabré et al. (2001), while Gaussier (2001) focusses specifically on bilingual term extraction tools. Finally, both Jaekel (2000) and Warburton (2001) present case studies that describe how terminology tools have been successfully implemented in a professional setting.

Notes

1. Most term banks maintain a presence on the World Wide Web, which means that information on specific term banks can be obtained by entering the term bank name (e.g., *Eurodicautom*) into a search engine in order to retrieve the appropriate URL.
2. Readers who would like to find out more about the advantages and disadvantages of other methods of terminology management can refer to Austermühl (2001: 103–107).
3. For example, see Heid (2001) for a discussion on collocation extraction, Pearson (1998) for a discussion on retrieving defining expositives from corpora, Hamon and Nazarenko (2001) and Jacquemin (2001) for descriptions of techniques for identifying synonyms, and Condamines and Rebeyrolle (2001) and Meyer (2001) for information about retrieving conceptual relations.
4. For more information on LISA, OSCAR and TBX, consult the LISA webpage: www.lisa.org.

References

- Ahmad, Khurshid and Margaret Rogers (2001) “Corpus Linguistics and Terminology Extraction”, in Wright and Budin (2001), pages 725–760.

- Austermühl, Frank (2001) *Electronic Tools for Translators*. Manchester: St. Jerome Publishing.
- Bourigault, Didier, Christian Jacquemin and Marie-Claude L'Homme, eds (2001) *Recent Advances in Computational Terminology*, Amsterdam/Philadelphia: John Benjamins.
- Cabré, M. Teresa (1999) *Terminology: Theory, Methods and Applications* (translated by Janet Ann DeCesaris). Amsterdam/Philadelphia: John Benjamins.
- Cabré, M. Teresa, Rosa Estopà Bagot and Jordi Vivaldi Palatresi (2001) "Automatic Term Detection: A Review of Current Systems", in Bourigault et al. (2001), pages 53–87.
- Clark, Robert (1994) "Computer-Assisted Translation: The State of the Art", in Cay Dollerup and Annette Lindegaard (eds), *Teaching Translation and Interpreting 2: Insights, Aims, Visions*, Amsterdam/Philadelphia: John Benjamins, pages 301–308.
- Condamines, Anne and Josette Rebeyrolle (2001) "Searching for and Identifying Conceptual Relations via a Corpus-based Approach to a Terminological Knowledge Base (CTKB): Method and Results", in Bourigault et al. (2001), pages 127–148.
- Dubuc, Robert (1985) *Manuel pratique de terminologie* (2e édition). Montréal: Linguatech.
- Dubuc, Robert (1997) *Terminology: A Practical Approach* (translated and adapted by Elaine Kennedy). Brossard, Québec: Linguatech.
- Gaussier, Eric (2001) "General Considerations on Bilingual Terminology Extraction", in Bourigault et al. (2001), pages 167–183.
- Hamon, Thierry and Adeline Nazarenko (2001) "Detection of Synonymy Links between Terms", in Bourigault et al. (2001), pages 185–208.
- Heid, Ulrich (2001) "Collocations in Sublanguage Texts: Extraction from Corpora", in Wright and Budin (2001), pages 788–808.
- Jacquemin, Christian (2001) *Spotting and Discovering Terms through Natural Language Processing*. Cambridge, Mass.: MIT Press.
- Jaekel, Gary (2000) "Terminology Management at Ericsson", in Robert C. Sprung (ed.), *Translating into Success: Cutting-Edge Strategies for Going Multilingual in a Global Age*, Amsterdam/Philadelphia: John Benjamins, pages 159–171.
- Kageura, Kyo and Bin Umino (1996) "Methods of Automatic Term Recognition: A Review", *Terminology* 3, 259–290.
- Kenny, Dorothy (1999) "CAT Tools in an Academic Environment: What are They Good for?", *Target* 11, 65–82.
- Koch, Katharina (1995) "Machine Translation and Terminology Database — Uneasy Bedfellows?", in Petra Steffens (ed.), *Machine Translation and the Lexicon*, Berlin: Springer-Verlag, pages 131–140.
- Lauriston, Andy (1997) "Terminology and the Computer", in Dubuc (1997), pages 179–192.
- L'Homme, Marie-Claude (1999) *Initiation à la traductique*. Brossard, Québec: Linguatech.
- Melby, Alan, Klaus-Dirk Schmitz and Sue Ellen Wright (2001) "Terminology Interchange", in Wright and Budin (2001), pages 613–642.
- Meyer, Ingrid (2001) "Extracting Knowledge-Rich Contexts for Terminography: A Conceptual and Methodological Framework" in Bourigault et al. (2001), pages 279–302.
- O'Brien, Sharon (1998) "Practical Experience of Computer-Aided Translation Tools in the Localization Industry", in Lynne Bowker, Michael Cronin, Dorothy Kenny and Jenni-

- fer Pearson (eds), *Unity in Diversity? Current Trends in Translation Studies*, Manchester: St Jerome Publishing, pages 115–122.
- Pearson, Jennifer (1998) *Terms in Context*. Amsterdam/Philadelphia: John Benjamins.
- Rondeau, Guy (1984) *Introduction à la terminologie* (2e édition). Québec: Gaëtan Morin.
- Sager, Juan C. (1990) *A Practical Course in Terminology Processing*. Amsterdam/Philadelphia: John Benjamins.
- Schmitz, Klaus-Dirk (1996) “Terminology Management Systems”, in Rachel Owens (ed.), *The Translator’s Handbook (3rd edition)*, London: Aslib, pages 221–239.
- Schmitz, Klaus-Dirk (2001) “Criteria for Evaluating Terminology Database Management Programs”, in Wright and Budin (2001), pages 539–551.
- Strehlow, Richard A. (2001) “The Role of Terminology in Retrieving Information”, in Wright and Budin (2001), pages 426–441.
- Vasconcellos, Muriel (2001) “Terminology and Machine Translation”, in Wright and Budin (2001), pages 697–723.
- Warburton, Kara (2001) “Globalization and Terminology Management”, in Wright and Budin (2001), pages 677–696.
- Wright, Sue Ellen and Gerhard Budin, eds (1997) *Handbook of Terminology Management, Volume 1*. Amsterdam/Philadelphia: John Benjamins.
- Wright, Sue Ellen and Gerhard Budin, eds (2001) *Handbook of Terminology Management, Volume 2*. Amsterdam/Philadelphia: John Benjamins.

CHAPTER 5

Localisation and translation

Bert Esselink

L10nbridge, Amsterdam, Netherlands

1. Introduction

Localisation is all about customising things (user manuals for products, especially software, and the products themselves) for a “local” audience. Although much emphasis is placed on new developments in translation such as web globalisation, translation workflow automation, and machine translation, most localisation vendors still mostly deal with localisation projects the way they have been doing it for the past ten years.

This chapter provides the basics of localisation, introduces project components and a typical team and process, and describes how the localisation industry has evolved since the beginning of the 1980s.

2. Introduction and definitions

The word “localisation” is derived from the term “locale”, which is defined in many different ways, depending on the source. In the *Collins Cobuild Dictionary*, for example, *locale* is defined as “a small area, for example the place where something happens or where the action of a book or film is set”. The *Sun Solaris Operating System Manual* defines *locale* as “a collection of files, data, and sometimes code, that contains the information needed to adapt Solaris to local market needs”. Most sources, however, define *locale* as all characteristics of the combination of a language and a region or country. Specific to a programming context, a locale defines all regional standards supported by a software product, such as date/time formats, sorting standards, currencies, and character sets.

The Localisation Industry Standards Association (LISA) defines *localisation* as follows: “Localisation involves taking a product and making it linguistically, technically, and culturally appropriate to the target locale where it will be used and sold.”

Often, *localisation* is abbreviated as *L10n*, where 10 represents the number of letters between the *l* and *n*.

Making a product *linguistically* appropriate to a particular market basically means translating it, and making it *technically* appropriate means adjusting all product specifications to support standards in the target market. *Cultural* adaptations are modifications of the source text to reflect situations and examples common in the target market. For certain software applications, the addition of newly developed modules or features is necessary: just think of a spelling checker in a word processor.

2.1 Related terms

In publications discussing localisation, often the terms “internationalisation” and “globalisation” are also referenced. According to LISA, **internationalisation** is

...the process of generalising a product so that it can handle multiple languages and cultural conventions without the need for re-design. Internationalisation takes place at the level of program design and document development.

Internationalisation is a task for developers, who need to include support for all possible target markets in their products. In most cases, better internationalisation automatically results in a more efficient localisation process. For example, if a software developer has created a software package where the address format automatically changes depending on the country that is chosen, there is no need to localise the product to a particular target market that requires a specific address format. Another common example is the use of Unicode¹ for character support; if a software developer integrates Unicode support into the product from the very beginning, there will be no need to change it to accommodate different writing systems, for example the “double-byte” systems needed for languages such as Japanese and Chinese.

The combination of internationalisation, localisation, and all other issues related to selling products in an international market is called **globalisation**. LISA defines globalisation as follows:

Globalisation addresses the business issues associated with taking a product global. In the globalisation of high-tech products this involves integrating localisation throughout a company, after proper internationalisation and product design, as well as marketing, sales, and support in the world market.

The World Wide Web has made the step to globalisation possible for many companies because e-commerce solutions have made it easier to reach an international consumer base. Designing and maintaining web sites in multiple languages is called **web globalisation**.

2.2 Translation versus localisation

Differences between “translation” and “localisation” can categorised as follows:

- activities,
- complexity,
- adaptation level, and
- technology used.

2.2.1 Activities

Traditionally, translation is one of the activities in projects where material is transferred from one language into another. Other activities that can be distinguished in traditional translation projects include terminology research, editing, proofreading, and page layout.

In localisation, many more activities can be identified. Examples of activities in localisation that are not necessarily part of traditional translation are:

- multilingual project management,
- software and online help engineering and testing,
- conversion of translated documentation to other formats,
- translation memory alignment and management,
- multilingual product support, and
- translation strategy consulting.

Most large, multi-language localisation agencies focus on these additional activities and outsource core translation activities to freelance translators or single-language vendors. Typically, only final language quality assurance is performed in-house by these agencies.

2.2.2 Complexity

Compared to traditional translation projects, managing software or web localisation projects can be very complex. Localisation projects typically contain a large number of components, such as software, sample files, online help, online and printed documentation, collateral materials such as product boxes and disk labels, and multimedia demos. In most cases, translation starts before the source material is final, so in localisation projects the source files are updated several times during translation.

As volumes are usually very large and all components contain dependencies, managing localisation projects can be tricky. Large volumes and tight deadlines require teams of translators who all need to be reviewed carefully to maintain consistency. For example, when translator A translates the software user interface and translator B the online help files, all references to the running software trans-

lated by translator B in the online help must exactly match the software translations that translator A has chosen.

Planning localisation projects is also a complicated task because many tasks depend on completion of previous tasks. For example, screen captures of localised software to be included in the online help or documentation cannot be created until the localised software has been engineered and the user interface tested.

2.2.3 Adaptation level

In software localisation projects, all local characteristics of the target market need to be implemented in the final product. Examples of these characteristics are language, culture, and all types of regional standards such as character set, currency, default page sizes, address formats, custom calendars, and date/time formats. A truly localised product should not only be in the target language but should also use default settings for the target locale. So, for example, a product sold in Germany should automatically use A4 as default page size, support input and output of accented characters, and display amounts using euros.

Apart from technical adaptations to software code, often complete rewrites (sometimes called “transcreations”) of sample files or marketing material need to be done before the content is acceptable for a certain target locale.

2.2.4 Technology used

In software localisation, the integration of translation technology has always been more prominent than in traditional translation. Because of the nature of software products and web sites, which are highly repetitive and updated on a regular basis, re-use of existing translations has become a competitive advantage and the use of **translation memory** (TM) (see Chapter 3) a must. Most software products are updated at least once a year, and web sites are often updated on a daily basis. As a result, TM tools have been applied successfully for many years in the localisation industry.

Other examples of translation technology widely applied in the localisation industry are software localisation tools for software user interface translations, terminology extraction and management tools, computer-aided translation (CAT) and machine translation (MT) systems.

3. Project components

Traditionally, software localisation projects consist of the following components: software, online help, and documentation. Since the late 1990s, a lot of these components have been converted to some type of Web-based format.

Even though the software component of a project is usually the smallest component in number of words, it may require a lot of engineering and testing work. The online help is normally the largest component with a lot of repetitive text, and documentation is included in both printed and online form.

Depending on the setup and design of the web site, web content is converted into HTML or XML format (these are “mark-up languages”, formats that determine how the web page will appear) or into database exports.

3.1 Software

Software translation typically refers to translation of the user interface elements of a software application, such as menus, dialog boxes, and messages. Figure 1 shows a Swedish dialog box, which has been truly localised because only date and time formats used in Sweden are shown.

In dialog boxes, not only do all options need to be translated, but also often resizing of items is necessary because of space restrictions. For example, if the translation of a button does not fit in the available space, the button needs to be resized. In some cases, resizing is not possible, for example when all languages are using the same dialog box template or screen layout. In those cases translators need to either abbreviate their translations or find shorter synonyms.

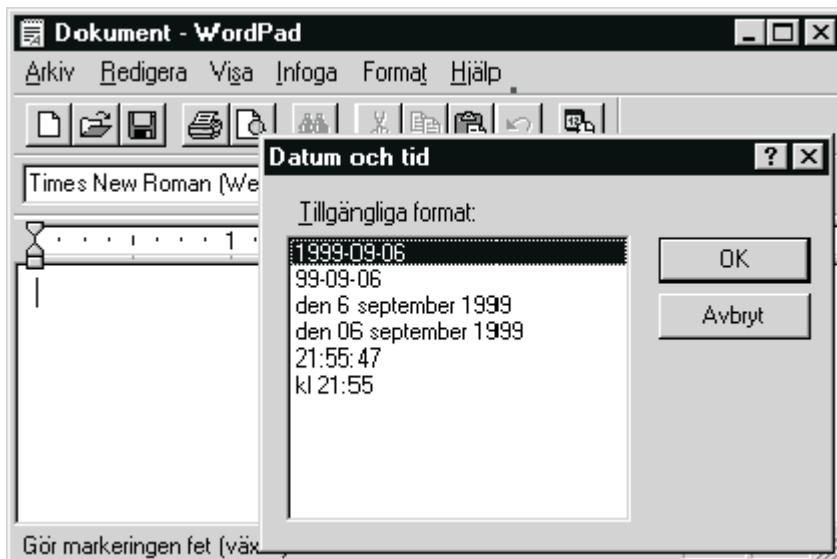


Figure 1. A dialog box localised for Swedish.

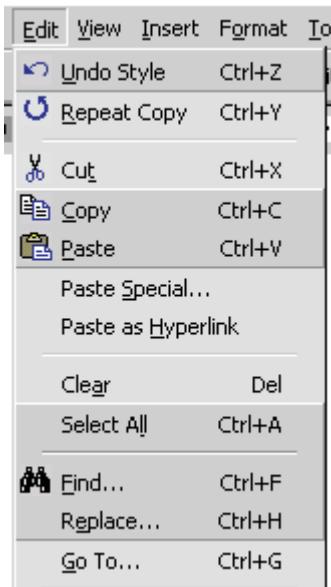


Figure 2. Drop-down menu showing hot keys.

Dialog boxes usually contain **hot keys**, as shown in Figure 2. These are the underlined letters that can be used in combination with the Alt key to access commands or options quickly. Each hot-key letter in a dialog box must be unique, so this is something that needs to be checked after translation. The same issue applies to menu items, which also usually contain hot-key letters.

Apart from the software application itself, software products often contain additional translatable components such as sample files, tutorials, and “wizards”. These often require extensive modifications to make them suitable for different target markets.

3.2 Online help

The online help is typically the largest component of a software localisation project. It contains the on-screen user assistance, which is usually context-sensitive and procedural. **Context-sensitive online help** means users can access help from any location in the application and the online help will automatically display help information which is relevant to that location, such as a dialog box.

Online help text tends to be very repetitive, which makes it the perfect candidate for translation with a TM tool. The elements of a typical online help file are the

table of contents, search keyword index, topic content with hyperlinks, and the full-text search index. All of these elements need to be translated, engineered, and tested before the localized product can be shipped.

Most of today's online help systems are based on the HTML file format. Online help files can be created by either assembling or compiling a set of HTML files and images.

Several tools have been developed to analyse and test localised online help, such as *HelpQA* and *HtmlQA* by SDL International.² These tools enable you to create detailed word counts and other statistics from help files, and after translation to compare the localised versions with the source material, both from a linguistic and a technical view.

3.3 Documentation

The number of printed documents included in software applications has gone down drastically over the past years. Most user assistance has gone online now, in the form of online manuals specifically aimed at printing or online help designed for online viewing.

The only printed manuals that often remain in software applications are an **installation guide** and guide for getting started. Other manuals, such as administrator or reference guides are included in an online format, such as HTML or *Adobe Acrobat PDF*.³

Software or hardware manuals are usually translated using a TM tool. Alternatively, many software publishers have developed conversion methods to convert online help information into online or printed documentation. For these conversions, tools like *Doc-to-help* are typically used. The most efficient way to publish multilingual information in different formats, e.g. online help, HTML, and PDF, is to use a single source publishing solution, for example based on a mark-up language like SGML or XML. This will allow publishers to create and translate information only once, and then publish it in different formats and layouts.

After translation, the layout of manuals is fixed, images are inserted, and the desired output created. Typically, localisation vendors will be asked to create PostScript files from localised manuals, which can then be directly processed by a printing firm.

Most software applications come with collateral material, for example a quick reference card, marketing material, the product box, and disk labels. These components are normally translated using the tool that was used to create the files, for example *QuarkXPress* for small documents, and *Adobe Illustrator* for disk labels.

3.4 World Wide Web

Increasingly, the World Wide Web is becoming multilingual. Companies who are already conducting business internationally have to localise their web sites; other companies see the Web as a perfect and easy means to start infiltrating foreign markets.

Most web sites contain a combination of marketing text, product information, and support information. Web sites can be static sites, with collections of HTML files, or dynamic sites, where information is stored in databases and XML or ASP pages are created on the fly with the appropriate text and images.

Localising a web site is in essence comparable to globalising an enterprise: for example, marketing material on the site cannot just be translated, but must be rewritten or adapted for target markets. Besides, the site should take into consideration the standards in all target countries where the company will do business, for example currency support, address and shipment details, local payment methods, local tax regulations, etc.

Especially in localisation of e-commerce sites, many issues besides the purely linguistic ones will need to be considered.

A typical characteristic of a corporate web site is the high frequency of changes and updates. In the case of multilingual sites, updates should ideally be released simultaneously in all languages of the site. The only possible way to achieve this is extensive automation of the translation workflow, for example by transferring changed web content automatically to a translator or translation vendor, and after review and approval automatically inserting the translations in the content database which publishes multilingual data to the web site.

Depending on the way companies approach web globalisation, localisation vendors will either receive batches of HTML files to translate and return, or database tables of information that will be used to generate web pages.

4. Project team

The team of people involved in a localisation project is a combination of people with management, linguistic, and technical backgrounds. A core production team usually consists of the following people:

- project manager,
- localisation engineer,
- language manager,
- translators,
- others.

In most cases, localisation vendors perform all non-translation activities in-house. All translation and proofreading work is outsourced.

4.1 Project manager

Project managers at localisation vendors oversee the entire project and maintain contact with all suppliers, team members, and the publisher. Their main responsibilities are:

- scheduling of all project activities,
- contact with supplier and client,
- resource and quality management,
- finances.

After project managers have created a project schedule, they oversee the project progress and make sure deadlines and project milestones are met. Project managers also maintain contact with suppliers, for example freelance translators, and with their clients. Contact with clients mainly consists of keeping them informed of the progress of the project, any issues, and finances. The financial responsibility of project managers consists of creating a project budget at project initiation, constantly comparing project costs to identify budget overruns, and invoicing customers for all services.

4.2 Localisation engineer

Localisation engineers are responsible for technical work that might be part of a localisation project. Examples of activities that localisation engineers perform are:

- project preparation,
- compiling software or online help,
- resizing dialog boxes,
- fixing localised layout before delivery.

Project preparation involves analysing the validity of the source material and creating kits that enable translators to start working immediately. Depending on the file format and development environment, localisation engineers compile software applications or online help projects from several source files into one binary file. With most localised software applications, there is a need to resize the screens to make translations fit, and in online help pages the layout needs to be fixed to make the localised text display correctly.

Localisation engineers do not need to be programmers or developers. Knowledge of how software applications and online help files are compiled and tested is

sufficient in most cases. However, this may change depending on the complexity of the software or help development format.

4.3 Language manager

In software localisation teams, one of the senior translators will act as a language manager for the project. Language managers are responsible for creating and maintaining language style guides, managing terminology to ensure consistency, reviewing the work produced by the translators, and answering questions raised by the translators regarding linguistic issues.

4.4 Others

Other people involved in localisation projects are, of course, the translators. Most localisation vendors use both in-house and freelance translators. Proofreaders are used to perform a final linguistic quality assessment of all translated material.

On the technical side, CAT software specialists prepare files for translation in a TM or software localisation tool, select the most appropriate tool to be used, and manage the TM databases.

Desktop publishing operators fix the layout of translated documentation files, create screen captures or edit images, and produce final PostScript files for printed documents or PDF files for online use.

5. Project process

A simplified localisation process contains the following steps:

- project setup,
- translation,
- review,
- production,
- quality assurance,
- project closure.

5.1 Project setup

Most localisation projects start with a kick-off meeting, where the publisher and localisation vendor meet to discuss the project plan. The vendor's project manager, lead translator, and technical manager typically attend kick-off meetings.

Also during the project setup phase, a list of commonly used terms in the

product is created and translated. This list will serve to ensure consistent use of terminology throughout the project. After the publisher has validated and approved the translated terminology list, translation of the source material can start.

The most important step in the project setup is analysis of the source material. The validity of the source files is tested: for example, does the software compile, does the manual contain all pictures, are instructions clear, etc. It is also during the analysis phase that word counts are generated and time estimates for the other project activities established.

Based on the information retrieved from the analysis, a project manager can create a project schedule.

After analysis the files can be prepared for translation. Engineers select the most appropriate CAT tools to be used for the project, create ready-to-start translation kits, and might even pre-translate part of the source material using existing glossaries or TM databases.

5.2 Translation

In localisation projects, software is normally translated first. As online help and documentation constantly refer to the software user interface, having a localised version of the software available prior to starting translation of the help files or documentation is advisable.

Depending on the file format, software resource files are translated using software localisation tools or TM tools. Software applications translated directly in the program files, such as .dll or .exe files, are typically localised using a software localisation tool, for example *Alchemy Catalyst*, *Passolo*, or *RC-WinTrans*.

As soon as a first draft translation of the main software user interface components is available — such as dialog boxes and menus — translation of the online help and translation can proceed. The software user interface terms are typically extracted to a **glossary**, which is then linked to the TM system used to translate the online help and documentation. For example, a user interface glossary in *Trados MultiTerm* will automatically display translations for the user interface terms while the online help text is being translated in *Translator's Workbench*.

At the same time, printed documentation such as a guide for getting started is translated; online manuals are often converted from the online help files after these have been localised and reviewed.

5.3 Review

If external suppliers or freelance translators have translated material, most localisation vendors schedule an in-house review of all translated material. During this

review, the translation quality and consistency is the main focus. For example, a software consistency check is performed, where reviewers verify whether localised software references in the online help or documentation match the actual localised user interface items.

At this point, many localisation vendors provide local representatives of the publisher with samples of the localised material for review and validation. This mainly serves to ensure that the publisher can identify any issues at an early stage so corrective action can be taken. The client validation process may be difficult to manage because most publishers assign local sales or marketing staff to perform this review, who may not necessarily have the time, bandwidth, or expertise to perform this review.

As soon as the translations have been reviewed and the quality assured, the files go into production. At this point, files are usually converted back from a TM tool to their original formats.

5.4 Production

An important production step is the compilation and engineering of the localised software application. Here, software localisation engineers perform tasks such as resizing of the dialog boxes to ensure translated options fit in the available space and checking for duplicate hot keys.

As soon as the user interface of the localised application has been fixed and validated, screen captures are created. **Screen captures** are images of user interface components, which are used in online help or documentation pages to clarify the information provided. Creation of screen captures often includes a certain amount of image editing to simulate particular situations or add localised sample text. Figures 1 and 2 in this chapter are examples of screen captures.

When the screen captures have been finalised, the online help component is compiled and tested. Engineers check the validity and layout of the localised files. Online help testing tools enable engineers to run checks automatically on the localised files and to compare the layout in two panels displaying the English and localised pages side by side.

Desktop publishers verify the layout and validity of manuals, generating indexes and inserting localised images and screen captures.

5.5 Quality assurance

After the online help has been engineered and tested and all images have been translated, all text is proofread to ensure the final linguistic quality. Proofreaders

often combine a language check with a layout check, assessing the final localised product like an end-user would.

Especially for printed documentation, this layout check is important because there may be errors in generated components, headers and footers, or page numbers.

When all proofreading corrections have been entered, the publisher receives (samples of) the localised product to perform an acceptance test. The result of an acceptance test is a *pass* or a *no-pass*, depending on the number and type of problems found. After a pass, the localised product is ready for delivery or hand-off to a printing firm.

5.6 Closure

This last phase of the project starts with delivery of the localised material to the publisher. Deliverables typically include the translated source materials, all compiled files, up-to-date TMs, and updated glossaries.

In the case of large projects, publishers may organise a wrap-up meeting, where the key team members meet to discuss and analyse the completed project and assess what could be improved in future projects.

After delivery, the localisation vendor archives all project materials to ensure that if updates to the product are to be localised in the future, all legacy material can quickly be located.

6. Translation technology

A distinction needs to be made between machine translation (MT) tools and computer-aided translation (CAT) tools. Where MT tries to replace a translator to a certain extent, CAT tools support the translator by preventing repetitive work, automating terminology lookup activities, and re-using previously translated texts. MT has not been applied much in the software localisation industry, mainly because, unlike in the automotive and aerospace industries, software publishers never really created their documentation in a structured way that would make MT successful. Although this seems to be gradually changing, the sections below will focus on CAT tools in order to reflect current practices in the localisation industry.

CAT tools, also called machine-aided translation tools, can be categorised as follows:

- translation memory (TM) tools,
- terminology tools,
- software localisation tools.

The first two types, TM and terminology tools, are typically combined for translation of documentation, online help, or HTML text. Software localisation tools are used to translate and test software user interfaces, i.e. dialog boxes, menus, and messages.

6.1 TM tools

Basically, a TM system is no more than a database which stores translated sentences (see Chapter 3). When a source text is imported into a TM tool, the text is segmented. Usually segmentation is performed on a sentence-by-sentence basis, where segments are separated by colons, commas, question marks, etc. However, it is also possible to segment texts on a paragraph basis, where segments are separated by paragraph marks. Each segment is a “record” in the TM database, and each record can store several fields, such as source-text segment, translated segment, language, name of translator, date of translation, type of text, and so on. The number of possible data fields in records varies per TM tool.

When text that has been segmented by a TM tool is translated, all translations are automatically stored in the records containing the source segments. If identical or similar sentences occur in the source text, the translations are automatically retrieved from the database and inserted into the target text. An identical segment that is automatically translated is called a **full match**; a similar sentence that is automatically translated is called a **fuzzy match**. Obviously, fuzzy matches need to be post-edited to make them correspond to the source text. A fuzzy match is, for example, a sentence where only one word has changed compared to an already translated sentence.

On large projects, TM databases can be shared amongst a team of translators. This means that if translator A has translated a sentence which also occurs in the text that translator B is translating, A’s translation will automatically be retrieved from the TM database and inserted in B’s target text.

Naturally, TM tools are particularly useful on large-volume texts, which contain a lot of repetitive text and where translations can be created on a one-to-one sentence basis. Using TM tools to translate marketing text or adverts is not often a good idea, simply because those types of texts often require many adjustments, rewrites, and other modifications.

In the software localisation industry, TM tools have always been very popular because of the short life cycle of software products. Most software products are updated at least once a year, and re-using translations of previous versions will increase time to market of localised versions drastically.

Examples of TM tools are Trados *Translator’s Workbench*, Atril *Déjà Vu* and STAR *Transit*.

6.2 Terminology tools

In localisation, terminology management is usually done in a very basic manner. Localisers typically do not create or use large multilingual terminology databases with term definitions, context, grammatical information, source, etc. (see Chapter 4). Instead, in most cases only **bilingual glossaries** of translated terms or phrases are used: for example, all translated terms from the software user interface. For this reason not only professional terminology management tools are used, but also basic glossary tools with limited search functionality.

Most TM tools run in conjunction with terminology management applications, which can be linked to the TM for **automatic terminology lookup**. Automatic terminology lookup means that terms in the source text, which are found in the dictionary or terminology database, are automatically displayed with their translations.

Examples of terminology tools are Trados *MultiTerm*, Atril *TermWatch*, and STAR *TermStar*.

6.3 Software localisation tools

Special tools have been developed to translate graphical user interfaces (GUIs) of software applications, i.e. the dialog boxes, menus, and messages that are displayed on a computer screen. These tools allow translators to view their translations in context: for example, translations can be entered directly in a dialog box and then saved.

Software localisation tools also contain features for automatically translating updated software with previously translated versions, and for running basic tests on localised software, for example checking if no translated text was truncated in the screens because of space restrictions.

Examples of software localisation tools are Corel *Alchemy Catalyst*, RC-WinTrans and *Passolo*, illustrated in Figure 3.

6.4 The next generation

Even though many translators still need to get acquainted with traditional translation technology such as TMs, the next generation of translation tools has already been introduced. Companies like Idiom and Trados offer automated Internet-based translation workflow solutions that automate many steps in translation projects. Texts are not only transferred automatically through each translation and review phase, but databases containing the source text are linked to translation technology that detects changed content and then first pre-translates it using a

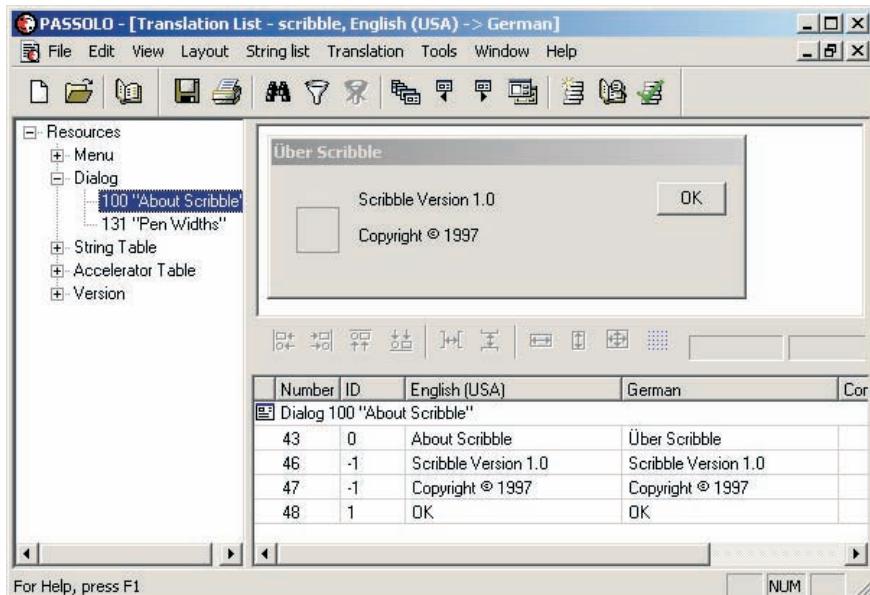


Figure 3. The *Passolo* software localisation system.

combination of TM and MT before it is forwarded to a human translator for post-editing.

These so-called “translation portals” and Internet-based “globalisation management systems” are mainly designed to deal with frequently changing content, such as text published on daily updated web sites.⁴

Technology and the Web will mean that translations will be done in a way totally different from how they were done for many centuries. Frequently updated content, geographically distributed resources, and pressure to keep prices down will result in further integration of technology and workflow automation in translation processes.

7. The localisation industry

In this section, we will focus on the localisation industry and introduce the history, major players and industry organisations.

7.1 History

Starting in the early 1980s, many software publishers realised they had to localise their products, mainly as a requirement to sell their products overseas. Before then, software was mainly published in the language the developers happened to speak. At that time, most large software publishers would either use individual freelance translators, single-language vendors, or in-house translation departments to perform the translation work. Smaller software publishers often requested translations from distributors or local sales people with no translation experience. As software publishers saw their in-house translation departments grow quickly through the large volumes of translatable text in software applications and documentation, most of them sought outsourcing possibilities in order to focus on their core business and keep the headcount down. Not only was the workload for internal translation departments very fragmented, but also project management was problematic, especially in projects involving dozens of languages.

The demand for outsourcing of translation activities combined with the large volumes and complexity of jobs automatically resulted in the launch of the first **multi-language vendors** (MLVs), who mainly focused on large-volume translation projects into multiple languages. MLVs also offered project management of these large, complex, and time-critical translation projects. MLVs were either start-ups, for example the INK network in Europe, or large divisions of established companies, such as Bowne's translation division, now called Bowne Global Solutions.

Still, many software publishers were experiencing bottlenecks just before their multilingual product releases, for example in their engineering and testing departments who suddenly found themselves having to test multiple language versions instead of just one English version. This called for an extended outsourcing model, which really took off in the beginning of the 1990s. Apart from translation services, MLVs also started offering engineering, testing, desktop publishing, printing, and support services.

This period can be considered as the start of localisation as we now know it. With teams of translators, project managers, engineers, testers, and desktop publishers, MLVs could provide one-stop multilingual solutions to software publishers.

An important trend that started taking shape in the late 1990s was the consolidation of the localisation industry. Many localisation vendors either merged with others or were acquired in order to achieve more market share, a better geographical spread, or more skills. In the 1990s, the number of large localisation vendors went down from 30 to 10. Examples of major consolidations taking place in the late 1990s and beginning 2000 were the acquisitions of Mendez first by Lernaut & Hauspie then by Bowne, Berlitz GlobalNET by Bowne, ILE/IC (INT'L.com) by L10nbridge, and ITP and ALPNET by SDL.

The yearly growth of the localisation industry has averaged 30% since the beginning of the 1990s. The most popular languages into which products are localised are French, Italian, German, Spanish,⁵ Brazilian Portuguese, and Japanese. In over 80% of all localization projects, the source language is English.

7.2 MLVs and SLVs

At the beginning of 2000, the major players in the localisation industry were Bowne Global Solutions, L10nbridge, and SDL.

These companies are all examples of MLVs offering a wide range of services, varying from e-services and testing (L10nbridge) to language training (Berlitz).

Although these MLVs usually get most publicity, most of the revenue in the translation and localisation industry is still generated by the thousands of **single-language vendors** (SLVs) and freelance translators that are active in every country. SLVs typically focus on one target language, have 1 to 30 employees, and offer mainly translation and desktop publishing services. Most SLVs work for MLVs; freelance translators usually work for both MLVs and SLVs.

7.3 Organisations

In 1990, the Localisation Industry Standards Association, LISA, was founded in Switzerland. LISA defines its mission as

... promoting the localisation and internationalisation industry and providing a mechanism and services to enable companies to exchange and share information on the development of processes, tools, technologies and business models connected with localisation, internationalisation and related topics.

LISA organises regular forums and workshops in which members can exchange information and attend seminars. These forums typically deal with business aspects of localisation and globalisation. Very little attention is paid to the activities and issues of translators.⁶

In Ireland, the Localisation Research Centre (LRC) was established at the University of Limerick as the result of a merger between the Centre for Language Engineering and the Localisation Resources Centre.⁷

7.4 Training and further reading

Not many opportunities exist for translators, engineers, and project managers to be trained in localisation processes and tools. Most localisation firms train their staff internally. Even standard technologies such as TMs are often not even covered in

translation or language studies.

Over the past few years, several surveys have been conducted to research how translation education could change curricula to train people better for the “real world” (see also Chapter 17). Examples of these surveys are:

- LEIT (LISA Education Initiative Taskforce), a commission that was formed in March 1998 and consists of representatives from universities in the USA and Europe.⁸
- LETRAC (Language Engineering for Translators’ Curricula), a project funded by the European Commission.⁹

Currently, more and more translation schools or language departments in universities specialise in localisation. There is a post-graduate course in localisation at the University of Limerick, and some institutes integrate localisation modules in translation education.

Especially for translators, not much information has been available on localisation. To fill this gap, *A Practical Guide to Localization* was written by the present author. The book was first published by John Benjamins in 1998 and the second edition published in the second half of 2000. Other books on software localisation and internationalisation are listed in the references, below.

7.5 Future developments

It is difficult to predict how the industry will develop in the future, especially because localisation is more fragmented than ever and everybody seems to be questioning what the localisation industry actually encompasses. Where localisation firms once distinguished themselves from traditional translation companies by specialising in translation, engineering and testing of software applications, now most of them are migrating to web localisation solutions. Since the Web is obviously not limited to software publishers only, many localisation firms find themselves again translating large-volume product and marketing information, which might have nothing to do with software applications, just like the good old days of translation.

In other words, it looks like the localisation industry will slowly be integrated back into the translation industry to form something most likely to be called the “multilingual publishing industry”. And when large localisation firms such as L10nbridge and Bowne Global Solutions keep moving upstream and offering content creation and product support solutions, the localisation industry of today will soon be called “multilingual solutions industry”.

Notes

1. Unicode is an internationally agreed standard for encoding different character sets in computers.
2. URL: www.sdlintl.com.
3. “Portable document format”: this is another format that determines the way documents appear on the computer: in this case, the text appears as a graphic image that looks like a printed page.
4. For more information on these types of translation technology, visit www.trados.com or www.idomtech.com.
5. These four languages are often referred to collectively as FIGS.
6. For more information on LISA, visit their Web site at www.lisa.org.
7. For more information on LRC visit their Web sites at lrc.csis.ul.ie.
8. For more information see www.lisa.org.
9. See www.iai.uni-sb.de/LETRAC.

References

- Esselink, Bert (2000) *A Practical Guide to Localization*. Amsterdam: John Benjamins.
- Luong, T. V., James S. H. Lok, David J. Taylor and Kevin Driscoll (1995) *Internationalization: Developing Software for Global Markets*. New York: John Wiley.
- Uren, Emmanuel, Robert Howard and Tiziana Perinotti (1993) *Software Internationalization and Localization*. New York: Van Nostrand Reinhold.

CHAPTER 6

Translation technologies and minority languages

Harold Somers
UMIST, Manchester, England

1. Introduction

In today's commercially-oriented world, much translation work is motivated by commercial considerations. Socio-economic factors thus influence the development of MT and CAT systems, and it is the major European languages (English, French, Spanish, German, Italian, Portuguese, Russian) plus Japanese, Chinese, Korean and to a certain extent Arabic that have received attention from the developers. But what if you work into (or out of) any of the several thousand other languages of the world? In this chapter we look at the case of MT and "minority" languages — an ironic term when one considers that the list of under-resourced languages includes several of the world's top 20 most spoken languages (Hindi, Bengali, Malay/Indonesian, Urdu, Punjabi, Telegu, Tamil, Marathi, Cantonese).

We have titled this chapter "Translation technologies and minority languages", since the minority languages are inferior in the provision of the whole range of computer aids for translators: not just MT systems, CAT systems, on-line dictionaries, thesauri, and so on, but even simple tools like spelling- and grammar-checkers. Because of accidents of world politics as much as anything else, the world's languages fall into three or four ranks, reflecting the computational resources available for them. This chapter will identify some languages which are more or less badly served (and other languages more usually designated as "minority" languages), and will briefly discuss what we can do about the situation.

2. Minority languages

The notion of "minority language" is relative, depending on the geographical standpoint of the observer. We can define the term from a Language Engineering

(LE) perspective (see below), or else from a local point of view. This latter option is relevant, since our proposed solution to the problem of linguistic knowledge acquisition relies on there being a community of professional linguists servicing the minority-language speaking community.

The UK is nominally an English-speaking country, with small regions where the indigenous Celtic languages are more or less widely spoken. However, a more realistic linguistic profile of the UK must take into account that there are significant groups of people speaking **non-indigenous minority languages** (NIMLs). Across the country, languages from the Indian subcontinent, as well as Cantonese, are widely spoken; other NIMLs are more regionally concentrated, e.g. according to the Commission for Racial Equality (1999), Greek and Turkish are among the 275 languages spoken in London. In other countries, the picture will be different, but only in the details.

While second- and third-generation immigrants are largely proficient in English, having received their schooling in this country, new immigrants as well as older members of the immigrant communities — especially women — are often functionally illiterate in English, even if they are long-term residents (Rudat, 1994). Many local councils, particularly in urban areas, recognize this, and maintain language departments to provide translation and interpreting services with in-house staff as well as lists of freelance translators. Their work includes translating information leaflets about community services, but also “one-off” jobs where individuals are involved, for example in court proceedings. Apart from serving the immigrant communities, refugees and, particularly in the major cities, asylum seekers, bring with them language needs that are being addressed by local government agencies. Just like translations in the private sector, “public service” translations come in all shapes and sizes. Some texts may amount to updates of previously translated material, may contain passages that are similar or identical to other texts that have already been translated, or may be internally quite repetitive.

Word-processing software is generally available for most of the world’s languages, at least as far as provision of **keyboards** and **fonts** for the writing system, allowing texts to be composed on a word-processor and printed, rather than handwritten. As we shall see, many of the other computational features associated with word-processing, that users of the world’s major languages are accustomed to, are simply not available for NIMLs, nor is there much evidence that the major providers of LE software will turn their attention towards NIMLs.

3. Computational resources for “exotic” languages

Language-relevant computational resources are certainly on the increase. The US-based magazine *Multilingual Communications & Technology* regularly lists new products and advances in existing products, and the software resources guide that it periodically includes grows bigger each issue. The translators’ magazine *Language International* has a similar “Language Technology” section. But just a glance at these publications reveals an overwhelming concentration on the few languages which are seen as important for world-wide trade: the major European languages (French, German, Spanish, Italian, Russian) plus Japanese, Chinese (i.e. Mandarin), Korean and, to a certain extent, Arabic. Their concern is the translation of documentation for products, commercial communications, and, especially recently, web-pages. Of course translation, like any other service industry, must be governed by market forces; but the languages that are of interest to commerce form an almost empty intersection with those of interest to government agencies dealing with the ethnic communities, refugees and asylum-seekers.

By way of illustration, we studied the WorldLanguage.com website — an online software and book shop — which extensively lists resources for a wide variety of languages. Table 1 shows the provision of translation-relevant LE resources for a selection of the NIMLs of significance in the UK.

We conducted a similar survey some years ago (see Somers, 1997, an early version of the present chapter), and while a lot of the white spaces in the corresponding table have been filled in in the intervening period, there are still significant gaps. For languages which use the Roman alphabet and a few diacritics, obvious non-language-specific provisions, such as keyboards, fonts and word-processors are available. Resources which in addition require word-lists (OCR and spell-checkers) are also now quite widely available. But products that involve more sophisticated linguistic content are largely absent. And if we look at languages that use a different writing system, we see more significant gaps.

A word of explanation is in order, regarding Table 1. First, note that this is just a snapshot, based on one albeit extensive listing. There may well be, for example, a Hindi translation product, but this resource did not, at the time of consulting (October 2002) list one. Second, no guarantee is made of the quality of the products listed. In particular, the dictionaries listed vary in size from a few thousand entries to more “serious” resources. The multilingual dictionaries are often simply word-lists with minimal coverage. The bilingual dictionaries are sometimes marketed as “translation” products, and some of the items that we have included here as translation products may be little more than automatic dictionaries.²

Table 1. Provision of computational resources for some “exotic” languages of relevance to the situation in the UK.

Language	Keyboard	Word-processor	Fonts	Desk-top publishing	OCR	Spell-checker	Grammar-checker	Dictionary (mono)	Dictionary (bi.)	Localisation tool	Translation	Speech products
Albanian		•	•	•	•	•						
Arabic	•	•	•	•	•	•		•		•	•	•
Bengali	•	•	•	•		•						
Bosnian												
Cantonese		•			•							
Chinese	•	•	•	•	•	•	•	•	•	•	•	•
Croatian	•	•	•	•	•	•	•	•	•	•	•	•
Farsi		•	•	•	•	•	•					
Greek	•	•	•	•	•	•		•	•	•	•	•
Gujerati	•	•	•	•	•	•			•			
Hindi												
Malayalam	•	•	•	•	•	•						
Marathi		•	•	•	•	•						
Polish	•	•	•	•	•	•	•	•	•	•	•	
Punjabi		•	•	•	•	•						•
Serbian		•	•	•	•	•	•		•			
Somali												
Sylheti												
Tamil	•	•	•	•		•						
Telugu	•	•	•	•		•						
Urdu		•	•	•	•	•						
Vietnamese		•	•	•	•	•		•		•		
Welsh	•	•	•	•	•	•			•			

Let us consider in a little more detail each of the categories listed in Table 1.

3.1 Keyboards

As mentioned above, provision of keyboards, word-processing and fonts is more or less trivial for languages using the Roman alphabet, though in some cases (e.g. Vietnamese) the requirement for unusual diacritics may be a challenge. Keyboard lay-out conventions differ from language to language, as Figure 1 illustrates, and

even amongst varieties of the same language: readers used to a US layout will notice some differences between the English keyboard illustrated in Figure 1 and what they are used to (location of “@”, “#” and “” symbols, and some keys differently located).

Keyboard software associates the appropriate character with the desired key-stroke, for example the top left-hand alphabetic key will give *a* if French keyboard is installed. Stick-on labels for the actual keys can also be acquired. For languages not using the Roman alphabet, bilingual keyboards are often used, as illustrated in Figure 2. But not all such languages are currently catered for.

For some languages, the writing system demands quite sophisticated input methods. This is true of several Indian languages, and also of Chinese and Japanese.

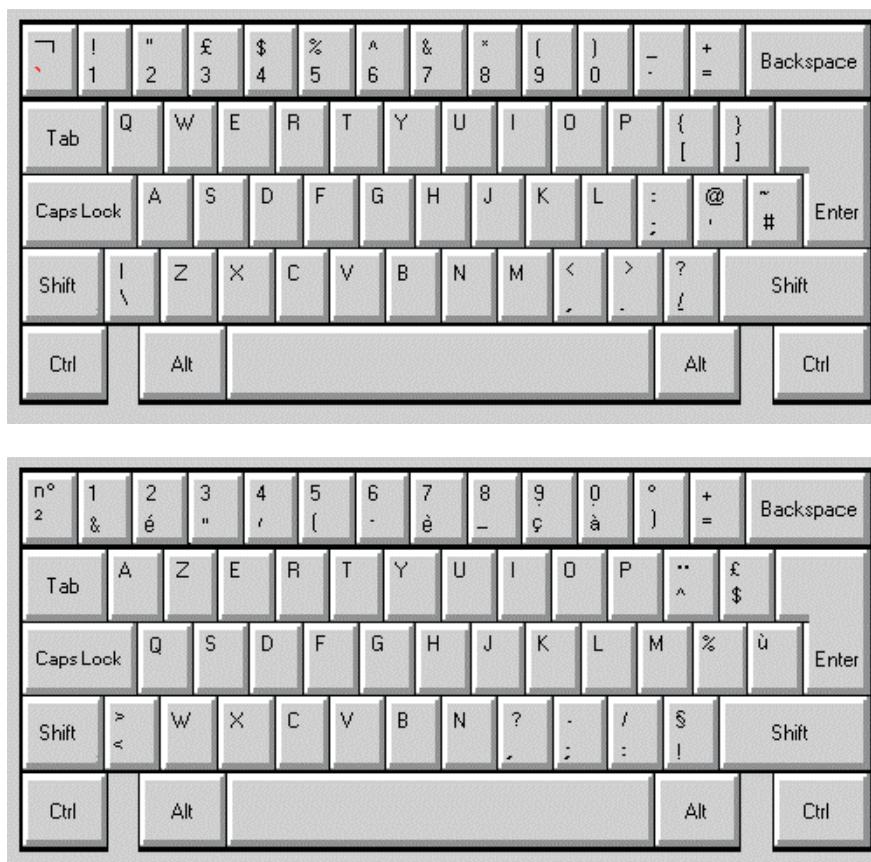


Figure 1. English QWERTY (above) and French AZERTY (below) keyboard layouts.



Figure 2. Arabic keyboard.³

Chinese is well provided for in terms of word-processing software; it should be noted however that software that goes beyond provision of character handling but is based on Mandarin may be unsuitable for Cantonese. For example, the typical input method for Chinese is based on pronunciation; but Cantonese and Mandarin (and other languages spoken in China⁴) have entirely different phonetic systems, and the “same” character is often pronounced quite differently.

3.2 Word-processing

As most readers will be aware, word-processing packages are much more than just computerised typewriters. Many of the tools that they include are language-sensitive, notably the following:

- Text justification
 - Automatic hyphenation programs
 - Auto-correction facilities (cf. spelling checkers, below)
 - Date and time stamps
 - Contents list and index creation for longer documents
 - Word counting

Text justification for languages which use a simple alphabetic writing system is relatively straightforward in principle — the program adds spaces between the words to make the line reach to the end — though even here there can be language-specific combinations. For example, some languages have a convention whereby

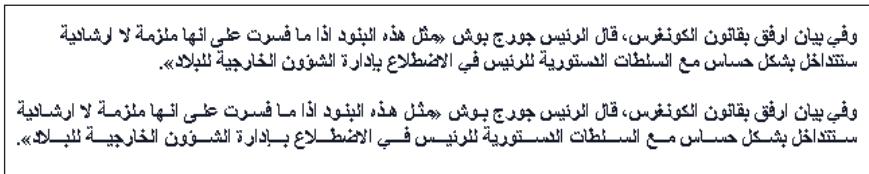


Figure 3. Justification in Arabic achieved by stretching the letter forms.

you can show that a word is stressed by adding spaces between the letters `l i k e t h i s`. The inserted spaces are not necessarily distributed evenly: you can generally have a bit more space immediately after a punctuation mark, especially a full stop (not to be confused with the mark of an abbreviation however, as in *Mr. Smith*). For languages with other writing systems, justification can be achieved in a number of other ways. In Arabic for example, some letter forms can be stretched to make the text fill the line, as shown in Figure 3, which shows the same short text unjustified (above) and justified (below).

Part and parcel of text justification is **hyphenation**. Most languages have quite strict rules about where words can possibly be hyphenated. These usually depend on phonetic and morphological structure, but they differ hugely from language to language (and even between varieties of the same language), and so must be especially provided for. English rules are mainly morphological, allowing *present-ation* while French rules more closely follow phonetics *présén-tation*. In French, double consonants can be split, e.g. *vil-lages*, whereas in English they must be kept together *villag-es*. Some rules change the spelling of the word: in Swedish and Norwegian, compounds which result in a sequence of three identical letters drop one of them as in *trafikkultur* (= *trafikk + kultur*). When hyphenated however, the missing letter is reinstated: *trafikk-kultur*. In German, at least before the recent spelling reforms, the character group *ck* enclosed by vowels could be hyphenated as *k-k*, e.g. *dicke* → *dik-ke*, though the correct hyphenation would now be *di-ce*. In Swiss German where the β-character of standard German is replaced by double *s*, words using this replacement are hyphenated between the double *s*, e.g. *Stras-se*, *grüs-se* unless the *ss* is the end of a compound element, e.g. *gross-artig* and not **gros-sartig*.

Auto-correction is when typographical mistakes are automatically corrected as you type them. Many word-processors have lists of words that are typically misspelled by users. These may include words which are commonly misspelled, such as **tommorrow*, **accomodation*, but also words where the keyboard layout leads to mistyping, such as **langauge*, **transalition*, **teh* (for *the*) and so on. Auto-correction tools often include automatic short cuts for special symbols so that typing *(tm)* gives the “™” symbol, or *1/2* the “½” symbol. Clearly, these are largely language-specific.

The way we write **dates** and **times** differs from language to language, a fact that the word-processor offering automatic date and time stamping must be aware of. My word processor allows me to insert today's date, *Thursday, 22nd August* with a single key click. But if I were typing in Malay, I would want it to insert *Khamis 22 Ogos*.

Automatic production of **contents lists** and **indexes** also need to be language-sensitive. Notably, alphabetisation differs in some languages: Danish has three letters *æ*, *ø* and *å* which are sorted at the end of the alphabet, after *z*, whereas alphabetical order in French and German (and others) takes no account of whether characters are accented. Languages like Spanish and Welsh have digraphs such as *ch*, *ll*, *rr* which are counted as separate letters. So for example *llamar* would come not between *línea* and *localidad*, but after *luz*.

Finally, **word-counting** facilities, much used by translators to calculate their fees, of course depend on an appropriate definition of what a “word” is.

In fact, at this level, provision for NIMLs is not too bad. Arabic word-processing packages can generally accommodate the different letter forms that printing requires, even for Urdu which has a number of extra letters customized from the Devanagari writing system used for Hindi — essentially the same language, though spoken by a different political and religious group — to cover Urdu sounds not found in Arabic. Even more “exotic” languages not listed in Table 1 are usually covered as far as fonts are concerned, and in the worst case the committed translator can get software for developing original fonts. It should not be forgotten however that high-quality systems for less popular languages are correspondingly more expensive, and may have less facilities and be harder to use than standard word-processing software.

3.3 OCR

Optical character recognition (**OCR**) is a process that converts scanned images into text. OCR is an important means of getting text into machine-readable form, which is essential if the translator wants to make use of it, for example to develop a translation memory, or to use as a resource for searching for terminology. When a page of text is scanned into a computer, it is stored as an electronic file made up of tiny dots, or pixels; it is not seen by the computer as text, but rather, as a “picture of text”. Early OCR technology in the late 1970s was very limited, and could only work with certain typefaces and sizes. These days, the software is far more advanced. Part of the way it works is by recognizing distinctive features in the shapes of the letters, but this process is backed up by language-specific knowledge about likely letter combinations, as well as knowing what the possible inventory of letter shapes is (e.g. whether to include accented characters, and if so, which ones). Thus, when you convert a scanned image into text, the OCR software needs to know what

language the text is written in. As Table 1 shows, while existing OCR software for the Roman alphabet can be easily adapted to new languages that use this alphabet, for other writing systems there is more work to be done. Despite the complexity of the writing system, OCR software for Chinese and Japanese is well advanced. But for Indian languages work has only just begun.

3.4 Spell-checking

Modern spell-checkers rely on a word-list (which is not the same as a dictionary, as it simply lists all the words, including their inflections, without distinguishing different word senses), as well as rules — or at least heuristics — for calculating the proposed corrections when a word is not found in the dictionary. Note that for some languages with agglutinative morphology, such as Turkish, where affixes can be stacked up potentially *ad infinitum*, it is effectively impossible to list all the possible word-forms. These heuristics may be based on the orthographic (and morphological) “rules” of the language concerned, or may take into account the physical layout of the keyboard. Alternatively (and more usually), they simply try a large number of permutations of the letters typed in, allowing also for insertions, deletions, substitutions and transpositions, and look these up in the word-list.

“Spelling” is in any case an notion that applies to alphabetic writing systems and is almost entirely meaningless for ideographic writing systems like Chinese and Japanese, and of arguable interpretation for syllabic or semi-syllabic writing systems. In addition, languages differ in the degree of proscription regarding spelling, especially for example in the case of transliterations of loan words or proper names. German newspapers in the 1980s varied between the phonetic *Gorbaschow* and the transliterated *Gorbaschev* spelling for the Soviet leader’s name. Hebrew is another good example: the normal writing system shows the consonants but not the vowels, so that מילון *mlwn* can be read as [milon] ‘dictionary’, [melon] ‘melon’ or [malon] ‘hotel’. However, for children and learners, the vowels can be shown as a system of diacritics above and below the letter, and sometimes there are variations in the spelling of a vocalized word and its unvocalized counterpart. There are other inconsistencies in the spelling of long vowels and the possible doubling of vowel letters, again especially in proper names: ‘Iraq’ can appear as any of ערך *irq*, עראק *iraq*, or עיראק *ijraq*. Hebrew dictionaries often compensate by listing a particular word under each possible spelling. Although the Academy of the Hebrew Language is the official body in Israel in charge of regulating spelling and other aspects of the Hebrew language, and has published official directives on spelling, Israeli publishers have, in many instances, found these to be a hindrance and have ignored the official rules.⁵

It is also interesting to consider what the purpose of a spell checker is. The following is the first verse of a humorous poem by Jerrold Zar (1994):⁶

I have a spelling chequer,
It came with my pea sea,
It plainly marques four my revue
Miss steaks eye cannot sea.

A spell-checker would of course fail to spot any of the spelling mistakes in the above text, since all of the misspelled words are homograph forms which in the correct context would be perfectly acceptable. For a spell-checker to correct this type of error would require sophisticated computational linguistics software that would analyse and in some sense “understand” the text it was checking. In fact, spell-checkers operate very simply: they search for “words” which are not in their word-list, and then permute the letters that are present in some more or less clever manner to suggest alternatives from the list of words that they *do* know about. So for example if I type *cardca*, my spell checker offers *circa*, *cardiac*, *cardkey*, *caracal*, and several others as possible corrections: no account is taken of the context, and the alternatives are simply words from the wordlist presented in an order determined by how different they are from what I typed in.⁷

3.5 Grammar checking

Style- or **grammar-checkers** at their best involve sophisticated computational linguistics software which will spot grammatical infelicities and even permit grammar-sensitive editing (e.g. search-and-replace which also changes grammatical agreement). In practise, “style-checking” tends to be little more than text-based statistics of average sentence length, word repetition, words and phrases marked as inappropriate (too colloquial), and use of certain words in certain positions (e.g. words marked as unsuitable for starting or ending sentences).

3.6 Dictionaries

As just mentioned, dictionaries are much more than word-lists: as well as distinguishing different word senses, they will usually offer some grammatical information. In one sense they are also something less than a word-list, since they usually do not list explicitly all the inflected or derived forms of the words. As Table 1 implies, it is useful to distinguish monolingual, bilingual and multilingual dictionaries. Although bilingual dictionaries are listed for many of the languages in Table 1, we should be aware that these are often very small (typically around 40,000 entries) and unsophisticated (just one translation given for each word).

In technical translation, whatever the field, consistency and accuracy of terminology is very important (see Chapter 4). Terminological thesauri have been developed for many of the “major” languages in a variety of fields with the aim of standardizing terminology, and providing a reference for translators and technical writers. A characteristic of NIMLs however is that they are often associated with less technologically developed nations, and so both the terminology itself and, it follows, collections of the terminology are simply not available. A similar problem arises from the use of a language in new cultural surroundings. For example, a leaflet explaining residents’ rights and obligations with respect to registering to vote or paying local taxes may not necessarily be very “technical” in some sense, but it will involve the translation of terminology relating to local laws which would certainly need to be standardized. If one thinks of the number of agencies involved in this type of translation — every (urban) borough or city council in the country, plus nationwide support agencies — then the danger of translators inventing conflicting terminology is obvious.

3.7 CAT and MT

After an initially disastrous launch in the 1980s, commercially viable CAT and MT software is now a reality: developers are more honest about its capabilities, and users are better informed about its applicability. But Table 1 shows only too clearly that this kind of software is simply not available for most of the languages we are interested in.

4. Developing New Language Engineering Resources

So what are the prospects of developing resources for these kinds of language and what steps can be taken to make available to translators of NIMLs some of the kinds of resources that translators working in the “major” languages are starting to take for granted? Current research in Computational Linguistics suggests some fruitful avenues. It is not appropriate to go into too much detail here, but the following sections will give a flavour of the prospects for future development.

4.1 Extracting monolingual word-lists from existing texts

From the point of view of the computer, fonts are simply surface representations of internal strings of character codes, so building up a dictionary of acceptable strings for a given language can be done independently of the writing system it uses. It is

not difficult (only time consuming) to take megabytes of correctly typed Hindi, say, and extract from it and sort into some useful order (e.g alphabetical order of the character codes) all the “words” that occur in the texts. Such a **corpus** of text could easily be collected by translators who work on a word-processor.

Assuming that spell-checking software is to some extent independent of the data (i.e. word lists) used, it should not be too difficult to develop **customized spell checkers**. Indeed, many word-processors permit the user to specify which word-lists or “dictionaries” are to be used, including the user’s own, and this can then be extended as it is used, by the normal procedure whereby users are allowed to add new words to their spell-checker’s word list.

4.2 Dictionaries and thesauri

Monolingual dictionaries, or thesauri (in the sense of lists of words organized according to similarity or relatedness of meaning) are a completely different matter. While the procedure described above could be used to generate a list of attested word forms, it is only the smallest first step towards developing a dictionary in the sense understood by humans. There is no obvious way to associate word meanings with different word-forms automatically. The best one could do would be to create and analyse “concordances” of the words (see Chapter 2.8), which would categorize them according to their immediate contexts, but this again is only a tool in the essentially human process of identifying word meanings and cataloguing them.

Of course, for many languages this has been done by lexicographers. Published dictionaries do exist for many of the languages we are interested in, and here there is a small glimmer of hope. Many dictionaries nowadays are computer-typeset so that **machine-readable dictionaries** are available, although they may include typesetting and printing codes and so on. Software that can extract from these the information that is needed for an on-line resource that is useful for translators has been widely reported.⁸ Unfortunately, this situation does not apply to all the languages we are interested in. For languages of the minority interest, dictionaries are often published only in the country where the language is spoken, where the publication methods are typically more old-fashioned, including traditional lead type-setting or even copying camera-ready type-written pages. To convert these into machine-readable form by using OCR implies a massive amount of work which is surely impractical, always assuming that OCR technology for the given writing system even exists (see above).

Another apparent source of data might be the World Wide Web. Unfortunately, again, this often turns out to be disappointing. For example, a search for “Urdu dictionary” brought up references to a number of websites replicating a very

small word list (1,900 entries), in transcription (not Urdu script) with very simple lists of English equivalents for the Urdu words. Another was similar in content, though a bit larger — “more than 11,000 words”. A third website claims to be an 80,000-word English–Urdu dictionary, where the Urdu words are shown as graphic images requiring a special plug-in.⁹ None of these really fit the bill.

4.3 Use of bilingual corpora

Like the (monolingual) corpus mentioned above, one could also envisage collecting samples of *pairs* of texts and their translations to create a **parallel bilingual corpus**,¹⁰ though in this case there would be the requirement that the original (source text) material was also in computer-readable format. There has been considerable research on extracting from such resources lexical, terminological and even syntactic information (Dagan and Church, 1994; Fung and McKeown, 1997; Gale and Church, 1991; van der Eijk, 1993). Before any information can be extracted from a bilingual corpus, the two texts must first be aligned. Of course this may be more or less trivial, depending on the language pair and the nature of the text. Again, much research has been done recently on this problem, much of it concerning corpora of related western languages, though a number of researchers have also looked at Chinese and Japanese. Fung and McKeown (1997) summarize the work done on this task. Of particular interest is work done on Chinese, where translations are rarely very “literal”, so that the parallel corpora are quite “noisy”. Fung and McKeown have developed a number of approaches to this particular problem.

One drawback is that even the best of these methods with the “cleanest” of corpora can only hope to extract much less than 50% of the vocabulary actually present in the particular corpus. With languages that are highly inflected, even this figure may be very optimistic. On the other hand, an aligned bilingual corpus presents an additional tool for the translator in the form of a translation memory. Even if this cannot be actually used by commercially available translation memory software, an aligned bilingual corpus can also be consulted on a word-by-word basis, where the translator wants to get some ideas of how a particular word or phrase has previously been translated (Isabelle and Warwick-Armstrong, 1993).

Besides extracting everyday bilingual vocabulary, attention has been focussed on identifying and collected technical vocabulary and terminology. Fung and McKeown (1997) describe how technical terms are extracted from their English–Chinese bilingual corpus. Dagan and Church (1994) describe a semi-automatic tool for constructing bilingual glossaries. Fung et al. (1996) show how the linguistic properties of certain languages can make this task more straightforward.

4.4 Developing linguistic descriptions

For most other purposes, a fuller linguistic description of the language is necessary. Sophisticated grammar checkers, and certainly CAT or MT tools, are usually based on some sort of linguistic rule-base. Although some work has been done on automatically extracting linguistic rules from corpora (Brent, 1993), nothing of a significant scale has been achieved without the help of a rule-based parser or an existing tree-bank. Two proposals directly related to developing MT systems for low-density languages describe software involving sophisticated interaction with a bilingual human expert (Jones and Havrila, 1998; Nirenburg and Raskin, 1998).

A more viable alternative might be to try to develop linguistic resources by adapting existing grammars. This might be particularly plausible where the new language belongs to the same language family as a more established language: a Bosnian grammar, for example, could perhaps be developed on the basis of Russian or Czech (cf. Hajič et al., 2000).

An alternative to full linguistic analysis is tagging. A tagged corpus is a useful resource, because it can be used to help linguists write the grammars that are needed for more sophisticated tools like MT. Tagging has the advantage of needing only a representative corpus with which to train the tagger. Researchers have generally reported a fairly clear correlation between the amount of text given as training data and the overall accuracy of the tagger, as might be expected. But this is a plausible route for developing sophisticated LE resources for NIMLs, always assuming that a linguist with the appropriate language background can be found to mark up the initial training corpus.

A final avenue that might be worth exploring is Example-based MT (EBMT), in its purest form requiring only a set of aligned previously translated segment pairs (Somers, 1999) (see Chapter 3.6).

5. Conclusions

This paper has discussed the grave lack of computational resources to aid translators working with NIMLs, and has attempted to identify some means by which this lack could be quickly addressed. The road will certainly be a long one, not least because the funding to support research in Computational Linguistics related to NIMLs will only come from government agencies, unless the private sector sees this as an area where it can make charitable donations. At least for the time being, there is no commercial interest in these languages. It is to be hoped that at least some of the lines of enquiry suggested here will prove fruitful in the short term.

Notes

1. Source: www.worldlanguage.com.
2. Indeed, the WorldLanguage.com website usefully draws attention to this: “There are several of types of Translation software utilities available. Included in these are interactive translation utilities that might be considered as ‘automatic’ dictionaries. In these utilities, the meaning or meanings of words or phrases are looked up automatically as the software moves through the text. Automatic or ‘Machine’ translation will go through the entire text or document, without stopping” (www.worldlanguage.com/ProductTypes/Translation.htm, emphasis original).
3. Source: www.savetz.com/vintagecomputers/arabic65xe.
4. To regard the various languages of China as “dialects” of a single “Chinese” language is linguistically unjustifiable. The languages spoken in this area are as (un)related as the languages of Europe; they happen to share a writing system, though this is only possible because it is based on meanings rather than pronunciations of words.
5. Source: The *About Hebrew* Internet site, hebrew.about.com/homework/hebrew/library/weekly/aa050800b.htm, and Arad (1991).
6. I am grateful to Ed Morrish for first drawing this to my attention. Zar’s poem, titled “Candidate for a Pullet Surprise”, is widely reproduced on the World Wide Web, often unattributed, or said to be of “unknown” authorship, and given the alternative title “An Owed to the Spell Chequer”. See also tenderbytes.net/rhymeworld/feeder/teacher/pullet.htm.
7. “Difference” is usually measured in terms on insertions, deletions, substitutions and transpositions: so *circa* is the best candidate because it involves only one substitution and one deletion; *cardiac* involves an insertion and a double transposition; and so on.
8. See for example Farwell et al. (1993). Mágan Muñoz (1998) discusses this tactic specifically for a minority language.
9. The three websites referred to are at www.rajiv.com/india/info/urdudic2.asp, www.ebazm.com/dictionary.htm, and urduseek.com/dict/. The larger site appears to have rather fewer entries than the 80,000 claimed, perhaps about 22,000.
10. Computational linguists use the word “parallel” to describe a corpus made up of texts in two (or more) languages which are translations of each other. Confusingly, the same term is used, for example in the field of Translation Studies, to refer to a collection of texts in different languages which are similar in content and form, though not necessarily mutual translations. Thus, a collection of recipes in different languages would be termed a “parallel corpus”. We will prefer the stricter interpretation whereby the texts are mutual translations.

References

- Arad, Iris (1991) *A Quasi-statistical Approach to Automatic Generation of Linguistic Knowledge*, PhD thesis, Department of Language and Linguistics, UMIST, Manchester.

- Brent, Michael R. (1993) "From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax", *Computational Linguistics* 19, 243–262.
- Commission for Racial Equality (1999) *Ethnic Minorities in Britain*. London: Commission for Racial Equality.
- Dagan, Ido and Kenneth Church (1994) "Termight: Identifying and Translating Technical Terminology", in *4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, pages 34–40.
- Farwell, David, Louise Guthrie and Yorick Wilks (1993) "Automatically Creating Lexical Entries for ULTRA, a Multilingual MT System", *Machine Translation* 8, 127–145.
- Fung, Pascale, Min-yen Kan and Yurie Horita (1996) "Extracting Japanese Domain and Technical Terms is Relatively Easy", in *NeMLaP2: Proceedings of the Second International Conference on New Methods in Language Processing*, Ankara, Turkey, pages 148–159.
- Fung, Pascale and Kathleen McKeown (1997) "A Technical Word- and Term-Translation Aid Using Noisy Parallel Corpora across Language Groups", *Machine Translation* 12, 53–87.
- Gale, William A. and Kenneth W. Chruch (1991) "Identifying Word Correspondences in Parallel Text", in *[DARPA] Workshop on Speech and Natural Language*, Asilomar, Calif, pages 152–157.
- Hajič, Jan, Jan Hric and Vladislav Kuboň (2000) "Machine Translation of Very Close Languages", *6th Applied Natural Language Processing Conference*, Seattle, Washington, pages 7–12.
- Isabelle, Pierre and Susan Warwick-Armstrong (1993) "Les corpus bilingues: une nouvelle ressource pour le traducteur", in Pierrette Bouillon and André Clas (eds) *La Traductique: Études et recherche de traduction par ordinateur*, Montréal: Les Presses de l'Université de Montréal, pages 288–306.
- Jones, Douglas and Rick Havrila (1998) "Twisted Pair Grammar: Support for Rapid Development of Machine Translation for Low Density Languages", in David Farwell, Laurie Gerber and Eduard Hovy (eds) *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas AMTA '98*, Berlin: Springer, pages 318–332.
- Mágan Muñoz, Fernando (1998) "Towards the Creation of New Galician Language Resources: From a Printed Dictionary to the Galician WordNet", in *Proceedings of the Workshop on Language Resources for European Minority Languages, First International Conference on Language Resources and Evaluation (LREC '98)*, Granada, Spain.
- Nirenburg, Sergei and Viktor Raskin (1998) "Universal Grammar and Lexis for Quick Ramp-up of MT Systems", in *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, pages 975–979.
- Rudat, Kai (1994) *Black and Minority Ethnic Groups in England*. London: Health Education Authority.
- Somers, Harold (1997) "Machine Translation and Minority Languages", in *Translating and the Computer 19: Papers from the Aslib conference*, London, pages not numbered. Available at www.ccl.umist.ac.uk/staff/Harold/aslib.ps.

- Somers, Harold (1999) “Review Article: Example-based Machine Translation”, *Machine Translation* 14, 113–158.
- van der Eijk, Pim (1993) “Automating the Acquisition of Bilingual Terminology”, in *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, The Netherlands, pages 113–119.
- Zar, Jerrold H. (1994) “Candidate for a Pullet Surprise”, *Journal of Irreproducible Results*, Jan/Feb, page 13.

CHAPTER 7

Corpora and the translator

Sara Laviosa

Università degli Studi di Bari, Italy

1. Introduction

In linguistics and related fields, a collection of texts stored on a computer, sometimes analysed automatically or semi-automatically, is known as a **corpus**. Until fairly recently the use of corpora has influenced the work of translators in an indirect way, particularly in the fields of terminology and translation aids. Terminology compilation, for example, is largely based on the linguistic and statistical analysis of representative corpora.¹ Moreover, corpus-based research has enhanced monolingual and bilingual dictionaries, and given rise to new reference tools such as “bridge bilingual dictionaries” (dictionaries of a language L_1 in which the definitions are translated into language L_2 , but the lay-out and head-words are those of the L_1) and dictionaries of collocations (words which occur together significantly). Computerised systems such as translation memories and example-based and hybrid MT programs have also been enhanced by the statistical and lexico-grammatical analysis of corpora (see Chapter 3).

In the last five years or so, there have been new and promising developments in the use of corpora. Translation theorists, for example, have begun to exploit corpora of original and translated text as a fruitful resource for the systematic study of the product and the process of translation. Contrastive linguists have recognised the value of translation corpora as resources for the study of languages, and translator trainers have designed general and specialised corpora to aid the comprehension of the source-language text and to improve production skills. Moreover, professional translators are becoming increasingly aware that the automatic analysis of language samples can assist them in many ways, for example in the interpretation of a literary source text, the retrieval of the linguistic context in which particular words are used in the target and/or source language, and in the comparative analysis of previous translations of the same original.

The aim of this chapter is to outline what can be regarded as some of the main

current and potential uses of corpora in the empirical study of translation, translator training, and professional translating.

2. A corpus typology for translation studies

Two main definitions of the term “corpus” have been put forward within **corpus linguistics**, the branch of descriptive linguistics that studies language on the basis of corpora. According to John Sinclair, a corpus is

... a collection of texts assumed to be representative of a given language, dialect or other subset of a language, to be used for linguistic analysis. (Sinclair, 1992: 2)

In the course of the EAGLES project, the following definition has been proposed:

... a collection of pieces of language that are selected and ordered according to explicit linguistic criteria in order to be used as a sample of the language. (EAGLES 1996)

Both definitions express an important feature of a corpus, namely that it is a **collection** of texts, either full running texts or text extracts. To a large extent this characteristic also applies to corpora in translation studies. However, as will be shown later, a corpus designed for translation purposes can consist of only two works, for example a source-language text and a target-language text.

Different types of corpus are being compiled for the study of translation and translating, translator training, and contrastive linguistics, as shown in Figure 1.

A **bilingual mono-directional parallel corpus** consists of one or more texts in language A and its/their translation(s) in language B, while a **bi-directional parallel corpus** consists also of one or more texts in language B and its/their translation(s) in language A. A **bilingual comparable corpus** consists of two collections of original texts in language A and language B. The two collections are generally similar with regard to text genre, topic, time span, and communicative function. A **monolingual**

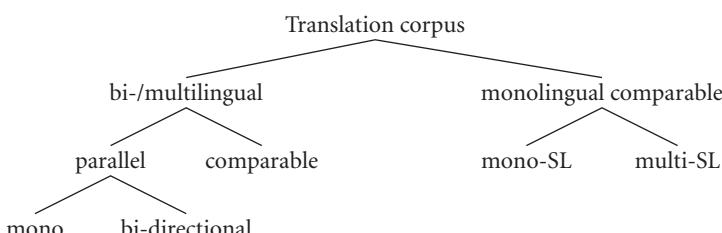


Figure 1. Types of translation corpus.

comparable corpus consists of two collections of texts in one language. One collection is made up of translations from one source language (mono-SL) or a variety of source languages (multi-SL), the other consists of original texts of similar composition to the translational component.

3. Descriptive corpus-based research in translation

Perhaps one of the first corpus-based descriptive studies of translation was Al-Shabab's (1996) investigation of vocabulary variety in a corpus of Arabic–English translations *vis-à-vis* original English texts. Al-Shabab hypothesised that there may be a difference in the type–token ratio² of translated texts and comparable target-language texts and that this discrepancy, if found, may be considered a characteristic of the language of translation.

He then proceeded to test this hypothesis on three corpora of radio news broadcasts in English: English broadcasts for Damascus English Service, based on Arabic originals; original English broadcasts for BBC Radio Four (a radio station which addresses an audience of native speakers); and original English broadcasts from the BBC World Service (a station which addresses a large audience of non-native speakers). The results showed that the translational group had a lower type–token ratio than the two original target-language corpora. Moreover, there were fewer cases of *hapax legomena*,³ and greater repetition of frequent words. Al-Shabab regarded these characteristics as three related aspects of **simplification** in the language of translation. However, he recognised that his findings may also be due to the direction of the translation process, which, in this case, was into English as a foreign, rather than a first language.

A different type of empirical work is Munday's (1998) analysis of shifts in Edith Grossman's translation, *Seventeen Poisoned Englishmen*, of a Spanish novel *Diecisiete ingleses envenenados* by Gabriel García Márquez. Munday made use of a variety of basic tools of corpus linguistics, for example, word-frequency lists, text statistics, and concordances,⁴ as aids to the inductive exploration of texts. Word-frequency lists were first obtained for both source and target texts and then compared for “spotting useful areas of investigation”. He used “intercalated text”, that is a text obtained by manually keying in the translation between the lines of the source text. He subsequently ran concordances of this intercalated text and used them to carry out a contextualised comparative analysis of all the instances of selected lexical items in order to examine some of the shifts “that build up cumulatively over a whole text”. This type of analysis is performed not to evaluate the quality of a given translation, but to understand the decision process underlying the

product of translation and to infer from it the translational norms adopted by the translator.

Munday's preliminary study of the first 800 words of his full-text parallel corpus revealed the existence of shifts in cohesion and word order which occur over the whole translation and have the effect of moving the narrative viewpoint from the first to the third person and so distancing the reader from the thoughts, experiences and feelings of the main character in the story.

Another corpus-based study of translation shifts is N. Scott's (1996) analysis of the novel *A Hora da Estrela* by Clarice Lispector, translated as *The Hour of the Star* from Portuguese by Giovanni Pontiero. Scott's aim was to look at **normalisation**, a term used to refer to "the translator's sometimes conscious, sometimes unconscious rendering of idiosyncratic text features in such a way as to make them conform to the form and norm of the target language and culture". The choice of the text was partly determined by the knowledge that the writer uses a peculiar style of writing, which exploits to the maximum the possibilities offered by vocabulary and syntax to express the uncertainty of her characters' thoughts, and partly by Scott's personal impression that the translation was "an easier text to grasp and follow in English". The methodology adopted was based on the use of a suite of computer tools provided by *WordSmith Tools* (M. Scott, 1996), and consisted of plotting and comparing the changes implemented by the translator *vis-à-vis* the source text. N. Scott examined, in particular, how the simple repetition pattern of the negative word *nao*, which is uniformly dispersed throughout the original text, is rendered in the English translation. She found that one Portuguese word *nao* had been translated into 72 different English words and, most significantly, it had been omitted 50 times. With the aid of a text aligner, Scott looked first at how each occurrence of *nao* had been translated and then grouped the translator's choices into seven categories which were ordered in a scale ranging from the most negative words (for example, *not*, *n't*) to omissions. The extremities of this scale represent the two poles of normalisation: normalisation due to the systemic constraints of the target language and normalisation resulting from the translator's own preferences. Scott concluded that the translator's choices, conscious or not, obligatory or optional, cause the breaking up of the cumulative effect of repetition of a single word *nao* and the end result is that "the nothingness conjured up in the source text has been weakened and dispersed".

Munday's and Scott's innovative works show how the analytical tools of corpus linguistics can be used heuristically to discover patterns that cannot be discerned through manual analysis, and to assess the cumulative impact that the individual choices of the translator have over the entire text.

The present author (Laviosa, 1998a) used a different type of corpus for studying the linguistic nature of English translated text. This corpus consisted of a subsection

of the English Comparable Corpus (ECC) (Laviosa-Braithwaite, 1996). It comprised two collections of narrative prose in English: one made up of translations from a variety of source languages, the other including original English texts produced during a similar time span. The study revealed four patterns of lexical use in translated versus original texts: relatively lower proportion of lexical words versus grammatical words, relatively higher proportion of high-frequency versus low-frequency words, relatively greater repetition of the most frequent words, and less variety in the words most frequently used. The author proposed to call these regular aspects of English translated text “core patterns of lexical use” in an attempt to convey the fact that, given that they occur in both the newspaper and the narrative prose subcorpora of the ECC, they may prove typical of translational English in general.

The unveiling of the specificity of the language of translation regardless of the contrastive differences between source and target language is also one of the principal aims of Øverås's (1998) investigation of **explicitation** (the extent to which an author makes things explicit in a text), expressed in terms of a rise in the level of cohesion in translational English and translational Norwegian. For her study Øverås used two subcorpora consisting of English and Norwegian translations of fiction, taken from the bi-directional English Norwegian Parallel Corpus (ENPC) compiled in the Department of English and American Studies of the University of Oslo under the direction of Stig Johansson. Øverås's comparison of the distribution of explicating and implicating shifts in the two corpora reveals a general tendency to explicitate in both translational English and translational Norwegian, notwithstanding a lower level of explicitation in Norwegian–English translations.

Øverås's and Laviosa's studies give an insight into the nature of translational language, traditionally described in the literature as “a third code” (Frawley 1984: 168), that is a unique language resulting from the confrontation of the source and the target codes, “a kind of compromise between the norms or patterns of the source language and those of the target language” (Baker, 2000a).

From the perspective of contrastive linguistics, Maia (1998) analysed the frequency and nature of the SVO sentence structure⁵ in English and Portuguese, particularly in those cases where the subject is realised by the first-person pronoun *I* and *eu* respectively, or by a name. The corpus analysed was a small bi-directional parallel corpus comprising a Portuguese novel and its English translation, and an English novel and its Portuguese translation. The texts contained a large number of monologues and dialogues, which were assumed to be representative of near speech-type usage. The parallel component of the corpus was regarded as appropriate for comparing how the same situation is represented in the two languages, while the original texts permit additional comparisons between the original languages on the one hand and between the translational and non-translational variety of the

same language on the other. The discrepancies observed in the frequency of personal subjects (realised by either names or pronouns) suggested, contrary to what happens in English, that the apparently subjectless V+O sentence structure is the norm, rather than the exception, in original Portuguese and that translational Portuguese is influenced by the norms of the English language. Moreover, while the use of *I* is syntactically necessary in English, the occurrence of the Portuguese equivalent *eu* seems to be related to pragmatic factors, such as thematisation, topicalisation and emphasis. On the basis of these findings, the author argued that “the flexibility of word order and the wider variation of thematisation in Portuguese in relation to English do at least allow for more subtlety in communication”.

Like Maia, Ebeling (1998) regarded parallel corpora as suitable sources of data for investigating the differences and similarities between languages, and adopted the notion of translation equivalence as a methodology for contrastive analysis. Ebeling used the ENPC to examine the behaviour of “presentative” English *there*-constructions⁶ as well as the Norwegian equivalent *det*-constructions in original and translated English, and original and translated Norwegian respectively. The corpus of original English revealed that *be* (and its variants) is by far the most common verb occurring in these structures, while Norwegian allows a much wider set of verbs, some in the passive voice. Ebeling’s analysis of the Norwegian translation equivalents of the English *there+be* constructions revealed an optional choice of specification with *det*-constructions containing verbs other than those of existence, *have*-existentials, *det*-constructions with passives. On the other hand, the English translation of *det*-constructions with active lexical verbs often leads to despecification, even more so than does the translation of *det*-constructions with passive verbs. These results partly confirmed the predictions put forward on the basis of the evidence from the original corpora and threw new light on the assumed relationship of equivalence between two structures found in English and Norwegian.

Johansson’s (1997) work was based on the extended ENPC, which is being expanded to include translations from English into Dutch, Finnish, German, Portuguese, and Swedish. Johansson carried out a three-way quantitative and qualitative comparison of the subject forms of the generic person in English (*one*), German (*man*), and Norwegian (*man*) using a multilingual mono-directional parallel corpus consisting of English originals and their translations into German and Norwegian. This study reveals that the English *one* is less frequent than the Norwegian *man*. This is, in turn, less frequent than the German *man*. The two most common sources of Norwegian *man* are the English *one* and *you* as well as a variety of non-finite constructions, which are rendered as finite ones in the translation. The English sources of German *man* are more varied: in addition to *one*, *you* and non-finite constructions, there are imperatives, and passive, dual-role or inanimate active subjects. With regard to differences in use of these forms, Johansson observed that

German *man* is stylistically neutral and corresponds to both the formal *one* and the informal *you* in English. While English *you* is the most common way of referring to people in general, in Norwegian both *man* and *du* (the second-person singular pronoun) are used.

Another parallel project currently in progress is INTERSECT (International Sample of English Contrastive Texts), which started in 1994 at Brighton University (Salkie, 1995, 1997). The aim of this initiative is to compile, align and analyse an English–French corpus of written texts, selected from a variety of genres, in order to investigate the changes that occur during the translation process and to test hypotheses derived from contrastive linguistics. One of the studies based on this corpus has shown that the English equivalents of the French *dont* — which, according to dictionaries, corresponds to *of which* or *whose* — reveal an unexpected variety of expressions compared with both the information given by bilingual dictionaries and the findings of small-scale contrastive studies that have looked specifically at translation strategies. Other contrastive studies have focused on the use of *but* and *mais*.

Finally, Geoffroy-Skuce (1997) analysed the functional ambiguity of the polysemous legal English adjective *adverse* in the compound-like modifier+head structure: *adverse effect*. Her study was based on a set of parallel concordances selected from a 2.5 million-word corpus of original court reports in Canadian English and their translations into Canadian French in the field of civil rights and criminal law from 1993 to 1994. The analysis was carried out on the basis of a Hallidayan theoretical framework of the “ideational” function⁷ of the clause where *adverse* occurs. The lexico-grammatical examination of the collocational context of each citation and the corresponding translation equivalent revealed that the meaning of *adverse* moves along a continuum stretching from the interpersonal to the experiential functions and that translation disambiguates the meaning of the source language with equivalents that may be semantically quasi-synonyms but at the same time functionally different and therefore not interchangeable. The findings have strong implications for specialised bilingual lexicography with regard to the actual information recorded in dictionaries and the representation of meaning. Geoffroy-Skuce pointed out, in fact, that the subtle, but common and significant functional variation of polysemous legal adjectives highlighted by her study is not accurately recorded in bilingual legal dictionaries where only collocations representing well-established legal concepts are featured. Moreover, the results suggest that, within legal discourse, English adjectives have field-specific conceptual centres and metaphoric extensions. On the basis of this, she put forward a case for a corpus-based computerised lexicographic representation of English legal adjectives based on a prototypical approach.

By adopting a genuine descriptive approach to translation while being, at the

same time, fully aware of the specificity of this act of language mediation, these contrastive linguists are able to break new ground in their respective research fields. They also demonstrate how fruitful and exciting the co-operation between contrastive linguistics, translation studies, and applied linguistics can be, through the adoption of a common corpus-based methodology, which has the potential for supplementing the information provided by general and specialised bilingual dictionaries as well as contrastive pedagogic and descriptive grammars.

4. Corpora in translator training

In the applied area of translator training, Zanettin (1998) and Gavioli (1996) demonstrate how small bilingual comparable corpora of either general (English and Italian newspaper articles) or specialised language (English and Italian medical reports) can be used to devise student-centred classroom activities involving first of all the creation of the corpora, then the analysis of individual words, discourse units and stretches of text. After taking part in these activities, students' translations carried out into and out of the mother tongue have revealed enhanced knowledge of the source language, clearer understanding of the source-language text, greater ability to produce fluent target-language texts, and, in the particular case of specialised translations, better understanding of the subject field. These encouraging results have given rise to the idea of providing each student of translation with a "translator trainee workstation" comprising a word processor, bilingual corpora and facilities for bilingual concordancing (Zanettin, 1998). Given the rapid growth in *Windows*-based text-retrieval software and Internet facilities, it is not unreasonable to predict that Zanettin's idea will fairly soon become a common feature in the more progressive and technologically advanced training institutions.

The European Union Lingua Project started in 1993 under the direction of Francine Roussel of the Université de Nancy II is another recent application of corpus-based research in translator training (King, 1997; Ulrych, 1997). The aim of the project is to create a multilingual parallel corpus as well as to develop, experiment with, and evaluate a *Windows*-based parallel concordancer, *Multiconcord* (King and Woolls, 1996; Johns, 1997; Woolls, 1997), which allows, among other functions, multiple-item searching with wild card, sorting and editing of citations, and the marking-up of text by the user. The texts included in the corpus are both original and translated works. The languages represented are Danish, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish, and Swedish. There is at least one source text in each of the ten languages so that no language is represented solely by its translational variety. The corpus can be processed by other types of software such as *WordSmith Tools*, which provides frequency lists, alpha-

betical lists, and collocational information.

This important resource has already been exploited by King (1997) and Ulrych (1997) in a variety of ways: to examine the translator's choices and strategies, compare translator behaviour *vis-à-vis* the information contained in bilingual dictionaries, test the validity of claims made in translation theory, and devise pedagogic materials for the training of translators and the teaching of foreign languages.

Still within a pedagogical perspective, Bowker (1998) carried out an interesting experiment in which she compared two translations produced by a group of native English translator trainees from French into English. One translation was carried out with the use of conventional reference material; the other with the aid of a specialised monolingual corpus on optical scanners, which was consulted with the analytical facilities provided by *WordSmith Tools*. The results revealed that the corpus-aided translations were of higher quality in respect of subject-field understanding, correct term choice, and idiomatic expression. Bowker observed that, although she did not find any improvement with regard to grammar or register, the use of corpora was not associated with poorer performance either. These interesting findings will no doubt inspire other scholars to pursue this experimental work not only in technical, but also general translator training.

5. A new type of corpus on the web

A new resource for the study of translation is the Translational English Corpus (TEC), a general-purpose, multi-source-language corpus of contemporary, written translational English. TEC was designed in 1994 in the (then) Department of Language Engineering at UMIST. It is now available on the World Wide Web for automatic analyses based on concordances and frequency lists.⁸

TEC consists of unabridged, published translations into English carried out by professional translators from European and non-European source languages. Four text categories are represented: biography, fiction, newspapers, and in-flight magazines. TEC is a new and versatile resource for the study of the lexical patterning of translational English independently of the influence of the source language. It has already supported research in the so-called "universals of translation", namely simplification (Laviosa-Braithwaite, 1996), normalisation (Kenny, 1999), and explicitation (Olohan and Baker, 2000). It has considerable potential for stimulating a variety of studies into the language of translation and for becoming an invaluable source of data for scholars, students, and practitioners working within translation studies. What follows are only a few suggestions about the possible lines of research that can be fruitfully pursued with TEC.

One may wish to compare texts translated by male versus female translators, or subcorpora of texts translated from different source languages. The language of translated newspaper articles can be compared with the language of translated narrative. The stylistic features of a particular translator could also be analysed by studying a representative sample of their translations (Baker, 1999, 2000b). Scholars interested in the study of shifts that occur during the translation process may want to analyse a parallel corpus which consists of a subcorpus of TEC texts translated from one source language on the one hand and their original texts on the other. TEC can also be combined with other corpora, for example one which consists of original English texts and has a similar composition to TEC. The resulting monolingual comparable corpus is particularly suitable for studying the typical linguistic features of translational versus non-translational English (Laviosa, 1997).

The practitioner too (instructor, assessor, translator) can benefit from the availability of a translation corpus. For example, with a parallel corpus based on one or more subcorpora of TEC, trainee translators can discover and discuss the regular solutions adopted by translators when they are faced with structural differences between source and target languages. An example of this type of application is described by Kohn (1996), who reports on the findings of a workshop on parallel concordancing, which analysed the original short story *Hundeblume* by Borchert and its Hungarian translation. The participants were able to discover that German compounds tended to be paraphrased in Hungarian with a present-participle construction.

The universal features of translational language can be studied with TEC-based parallel and monolingual comparable corpora. Their systematic investigation can be carried out either from a descriptive point of view to identify the typical pattern of the language of translation or with the aim of improving translators' performance. With a common corpus-based methodology the findings of descriptive and applied research will become reasonably comparable and a fruitful dialogue between the two branches of the discipline will not only be possible, but it will, in the long term, be perceived by both sides as being highly desirable for the mutual progress of their respective areas of interest.

Different types of questions can be asked by scholars who work in separate fields of translation studies, while the use of a common corpus-based methodology will enhance dialogue and exchange of data and results among them. This development, if forthcoming, will contribute to bringing unity to the discipline while maintaining its productive diversity.

6. Corpora and the professional translator

There are at least two ways in which the practising translator can benefit from the new developments in corpus-based research outlined in this chapter. They can draw on the insights provided by descriptive studies into the differences and similarities between languages, the strategies adopted by translators, the patterning of translational language independently of the influence of the source language, as well as the most common translation equivalents. These insights can not only enhance translation performance in terms of fluency and accuracy, but will enable them to refine their awareness of the nature of translation as a particular type of language mediation. On the other hand, the availability of user friendly and relatively inexpensive software for the automatic processing of texts as well as the accessibility of corpora on the World Wide Web may encourage translators to carry out their own linguistic, stylistic and textual analyses of single input texts or corpora for their individual needs. This will empower the translator, who will be in a position to integrate the skills and knowledge of the researcher and the practitioner and so be able to bridge the timely gap between scholarly and professional work.

Notes

1. Baker (1995); Bowker (1995a,b, 1996)
2. This measure of vocabulary richness in a text is the ratio of different words, or “types”, to the total number of words, “tokens”: the higher the ratio, the richer the vocabulary.
3. A type occurring just once in the entire text.
4. A “concordance” is a listing of all the occurrences of a given word, usually arranged so that similar contexts are juxtaposed, e.g. in alphabetical order of the succeeding word. See Kennedy (1998: 247ff). See also Chapter 2.8.
5. Subject (S) followed by verb (V) followed by object (O).
6. Such as *There was once a man who ...*
7. The “ideational” function serves for the expression of “content”: that is, of the speaker’s experience of the real world. (Halliday, 1970: 143).
8. <http://www.umist.ac.uk/ctis>

References

- Al-Shabab, O. S. (1996) *Interpretation and the Language of Translation: Creativity and Convention in Translation*. Edinburgh: Janus.

- Baker, Mona (1995) "Corpora in Translation Studies: An Overview and some Suggestions for Future Research", *Target* 7, 223–243.
- Baker, Mona (1999) "The Role of Corpora in Investigating the Linguistic Behaviour of Professional Translators" *International Journal of Corpus Linguistics* 3, 1–18.
- Baker, Mona (2000a) "Linguistic Perspectives on Translation", in Peter France (ed.) *The Oxford Guide to Literature in English Translation*, Oxford: Oxford University Press, pages 20–26.
- Baker, Mona (2000b) "Towards a Methodology for Investigating the Style of a Literary Translator", *Target* 12: 241–266.
- Bowker, Lynne (1995a) *A Multidimensional Approach to Classification in Terminology: Working Within a Computational Framework*, PhD thesis, UMIST, Manchester.
- Bowker, Lynne (1995b) "LSP Corpora in NLP: some Fundamentals and Approaches in the Discipline of Terminology", in *Proceedings of the 4th International Conference on the Cognitive Science of NLP*, Dublin, pages 1–9.
- Bowker, Lynne (1996) "Towards a Corpus-based Approach to Terminography", *Terminology* 3, 27–52.
- Bowker, Lynne (1998) "Using Specialized Monolingual Native-Language Corpora as a Translation Resource: A Pilot Study", in Laviosa (1998a), pages 631–651.
- EAGLES (1996) *Recommendations on Corpus Typology*. Pisa: ILC – CNR.
- Ebeling, Jarle (1998) "Contrastive Linguistics, Translation, and Parallel Corpora", in Laviosa (1998a), pages 602–615.
- Frawley, William (1984) "Prolegomenon to a Theory of Translation", in William Frawley (ed.), *Translation: Literary, Linguistic, and Philosophical Perspectives*, London: Associated University Presses, pages 159–175.
- Gavioli, Laura (1996) "Corpora and the Concordancer in Learning ESP. An Experiment in a Course for Interpreters and Translators", in G. Azzaro and M. Ulrych (eds.) *Lingue a Confronto. Atti del XVIII Convegno AIA, Genova, 30 Settembre – 2 Ottobre 1996, vol. II*. Trieste: EUT.
- Geoffroy-Skuce, Anne (1997) "Polysemous Adjectives in Legal Translation", in K. Simms (ed.), *Translating Sensitive Texts: Linguistic Aspects*, Amsterdam: Rodopi, pages 155–168.
- Halliday, M. A. K. (1970) "Language Structure and Language Function", in John Lyons (ed.) *New Horizons in Linguistics*, Harmondsworth: Penguin, pages 140–165.
- Johansson, Stig (1997) "Using the English–Norwegian Parallel Corpus — A Corpus for Contrastive Analysis and Translation Studies", in *Practical Applications in Language Corpora: The Proceedings of PALC97*, Łódź, Poland, pages 282–296.
- Johns, Tim (1997) Multiconcord: the Lingua Multilingual Parallel Concordancer for Windows, <http://web.bham.ac.uk/johnstf/lingua.htm>.
- Kennedy, Graeme (1998) *An Introduction to Corpus Linguistics*. London: Longman.
- Kenny, Dorothy (1999) *Norms and Creativity: Lexis in Translated Text*, PhD Thesis, UMIST, Manchester.
- King, Philip (1997) "Parallel Corpora for Translator Training", in *Practical Applications in Language Corpora: The Proceedings of PALC97*, Łódź, Poland, pages 393–402.
- King, Philip and David Woolls (1996) "Creating and Using a Multilingual Parallel Concordancer", in *Translation and Meaning, Part 4: Proceedings of the Łódź Session of the 2nd*

- International Maastricht–Łódź Duo Colloquium on Translation and Meaning*, Łódź, Poland, pages 459–466.
- Kohn, János (1996) “What Can (Corpus) Linguistics Do for Translation?”, in Kinga Klaudy, José Lambert and Anikó Sohár (eds), *Translation Studies in Hungary*, Budapest: Scholastica, pages 39–52.
- Laviosa, Sara (1997) “How Comparable Can ‘Comparable Corpora’ Be?”, *Target* 9, 289–319.
- Laviosa, Sara (1998a) “Core Patterns of Lexical Use in a Comparable Corpus of English Narrative Prose”, in Laviosa (1998b), pages 557–570.
- Laviosa, Sara (ed.) (1998b) “The Corpus-based Approach: A New Paradigm in Translation Studies” (Special edition), *Meta* 43.4.
- Laviosa-Braithwaite, Sara (1996) *The English Comparable Corpus (ECC): A Resource and a Methodology for the Empirical Study of Translation*, PhD thesis, UMIST, Manchester.
- Maia, Belinda (1998) “Word Order and the First Person Singular in Portuguese and English”, in Laviosa (1998a), pages 589–601.
- Munday, Jeremy (1998) “A Computer-Assisted Approach to the Analysis of Translation Shifts”, in Laviosa (1998b), pages 542–556.
- Olohan, Maeve and Mona Baker (2000). “Reporting *That* in Translated English: Evidence for Subconscious Processes of Explication?”, *Across Languages and Cultures* 1, 141–158.
- Øverås, Linn (1998) “In Search of the Third Code: An Investigation of Norms in Literary Translation”, in Laviosa (1998a), 571–588.
- Salkie, Raphael (1995) “Intersect: A Parallel Corpus Project at Brighton University”, *Computers & Texts* 9, 4–5.
- Salkie, Raphael (1997) “Naturalness and Contrastive Linguistics”, in *Practical Applications in Language Corpora: The Proceedings of PALC97*, Łódź, Poland, pages 297–312.
- Scott, Mike (1996) *WordSmith Tools: Software Language Tools for Windows*, Oxford: Oxford University Press.
- Scott, Nelia (1996) “Investigating Normalization in Literary Translation”, Paper presented at “Looking at Language into the Millennium” seminar, Department of English Language, University of Glasgow, 14 May 1996.
- Sinclair, John (1992) “Lexicographers’ Needs”, in *Workshop on Text Corpora*, Pisa, Italy, pages 1–4.
- Ulrych, Margherita (1997) “The Impact of Multilingual Parallel Concordancing on Translation”, in *Practical Applications in Language Corpora: The Proceedings of PALC97*, Łódź, Poland, pages 421–435.
- Woolls, David (1997) *Multiconcord: Software for Multilingual Parallel Concordancing*. Birmingham: CFL Software Development.
- Zanettin, Federico (1998) “Bilingual Comparable Corpora and the Training of Translators”, in Laviosa (1998a), pages 616–630.

CHAPTER 8

Why translation is difficult for computers

Doug Arnold

University of Essex, Colchester, England

Why is it difficult to get computers to translate? Our answer to this will be in two parts. The first part consists of some general remarks about the nature of translation, and the abilities of computers. These will lay out the ground and provide a general but rather unsatisfactory answer to the question. The second part will look in more detail at the sorts of problem that create the difficulty, and provide a more detailed and revealing answer.

1. Translation and computers

Part of the reason why translation is difficult for computers is that translation is just difficult: difficult even for humans. Translating is a many-faceted skill that goes well beyond mere competence in two languages. Roughly speaking, the job of a translator is take a text in one language (the source language) and produce a text in another language (the target language) which is in some sense equivalent. Before we talk about why this is difficult, we should notice that translators are often asked to do rather more than this. In particular they are often expected to produce a text that is in some sense “good” in its own right — clear, unambiguous, interesting, persuasive, elegant, poetic, gripping, etc., according to the kind of text being translated. While this is understandable, it is clearly somewhat unfair, especially when one is thinking about trying to automate the process. It is one thing to ask a computer to produce a target text which is (in some sense) equivalent to the source text; it is quite another to ask the computer to make it *interesting*. So, in asking why translation is difficult for computers, we should be careful to restrict ourselves to the translation job proper: to be concrete, let us imagine that anything the computer produces will be post-edited for qualities other than equivalence with the source text. All we want from the computer is some kind of **draft quality** translation: something which is more or less faithful to the original, understandable in its own

right, and which is a reasonable starting point for a polished translation.

Of course, this is still very difficult, even for a skilled human, because the appropriate notion of “equivalence” is difficult to pin down, and can vary greatly depending on the kind of text involved. For example, in translating texts for an online help system, the length of the source text (number of characters) may be important, since the translation may have to fit in the same area of screen as the source text. While one normally expects a translation to be roughly the same length as the original, one would not normally worry about counting characters. Let us try to ignore these complications also, and focus on cases of translation where the key point is just to convey the *content* of the source text.

Unfortunately, this is still a tall order, because languages do not always allow the same content to be expressed. There are many well-known cases where one language lacks a precise equivalent for a term in another. In English, one can be vague about the gender of a friend, without seeming evasive. This is harder in French, where one has a choice between terms for male *ami* and female *amie*. Conversely, it is hard in English to refer to a friend who is female without going too far (*girlfriend*) or seeming to labour the point (*female friend*). So let us be a little less ambitious, and ask for only *approximately* the same content.

Even so, translating is a difficult task. In particular, it is a *creative* task, for at least two reasons. First, translators are often expected to be able to coin translations of novel terms that appear in the source text. Second, translators are often required to act as cultural mediators, conveying to readers of the target language what may be obvious to readers of the source language. A very clear case of this occurs with the translation of religious texts (how should one translate *Man shall not live by bread alone* for readers for whom bread is an alien or exotic foodstuff?)

Computers are fundamentally just devices for following rules, mechanically and literally, albeit with considerable speed and precision. Rule following can produce a kind of creativity, but not the kind of creativity required for these tasks. Coining a new piece of terminology is more a matter of inventing a rule than following a rule, and cultural mediation requires very sophisticated reasoning: one must be able not only to extract the meaning from a text, but also be able to think about what meaning a potential reader would extract. To avoid these problems, we should restrict ourselves to cases where readers of source and target text can be regarded as sharing the same culture and background knowledge (e.g. by being members of the same profession or scientific discipline), and where problems of novel terminology either do not arise or can be solved by a human in interaction with the computer.

The translation task we have now is one of taking a text written in one language and producing a text in another language with the same approximate content, where readers of the target text are expected to share the same knowledge and culture as the

readers of the source text, where there are no problems due to new terminology, and where we expect a human translator to be involved in producing a polished result.¹ For the most part, the aim of MT research over the last forty or so years has been to automate this process. Despite considerable progress, despite the fact that the aim has actually been achieved for some languages, and some restricted domains and text types, it still poses fundamental practical and theoretical problems.

At the root of these problems are four particular limitations of computers, namely, the inability of computers to:

- (i) perform vaguely specified tasks
- (ii) learn things (as opposed to being told them)
- (iii) perform common-sense reasoning
- (iv) deal with some problems where there is a large number of potential solutions.

Precisely formulated rules are required because they must, ultimately, be interpreted in terms of the normal operations of computer hardware. Much of the difficulty of natural language processing in general, and MT in particular, arises from the difficulty of finding sufficiently precise formulations of intuitively very straightforward ideas like “in English, the subject usually comes before the verb” (the really problematic word here is *usually*, of course). Moreover, a precise formulation is not enough. There are problems for which rules can be formulated precisely, but for which solutions still cannot always be computed (any task that involves examining every member of an infinite set, for example).

Learning also poses fundamental problems from a computational perspective. There are several reasons for this, one of which is to do with the fact that it involves classification, which involves the notion of similarity, which is a vague notion, another being the fact that it involves genuine creativity (rule inventing, not rule following). There *are* learning algorithms for some tasks, but there is no general reliable procedure for learning the kinds of knowledge required for MT. In this area, what a computer needs to know, it must be told, in the form of explicit rules, written by humans.

The third problem is that computers cannot perform **common-sense reasoning**. There are several reasons for this, but perhaps the most serious is the fact that common-sense reasoning involves literally millions of facts about the world (water is wet, men don’t get pregnant, most people have two feet, sheep are larger than fountain pens, if B has been put in A then A contains B, for A to contain B, A must be larger than B, and so on). The task of coding up the vast amount of knowledge required is daunting. In practice, most of what we understand by “common-sense reasoning” is far beyond the reach of modern computers.

The fourth fundamental difficulty for computers arises even for precisely

specified problems which do not involve learning. It is the problem of **combinatorial explosion**. Suppose there are a number of slots each of which can be filled in one of two ways (say, by a 0 or a 1), and that we have to consider every way of filling the slots (the worst case). The number of possibilities very quickly becomes very big. There are two ways of filling one slot, four ways of filling two, and in general 2^n ways of filling n slots. Every time we add a slot, we double the number of possibilities, and hence the amount of time required. Suppose that it takes 1 millisecond to consider one solution: ten slots involves $2^{10} = 1024$ possibilities, requiring just over a second. With 20 slots, the number of possibilities rises to 1,048,576, requiring over two hours. With 30 slots, the time goes up to 12 days, with 40 it goes up to over 34 years. Dealing with 41 slots would take over 64 years, which is too long for most humans to wait. Improvements to computer hardware are insignificant in the face of this sort of problem: buying a computer which is twice as fast as your present one allows you to deal with exactly one more slot in any given time.

The bad news, from an MT perspective, is that each of these limitations is relevant. Thus, a general, though not very revealing answer to the question we started with would be: “Because it involves problems that resist an algorithmic solution, including common-sense reasoning, learning, and combinatorially explosive tasks”. In order to give a more systematic and revealing answer, we need to look at the various tasks involved in different approaches to MT.

There are three “classical” architectures for MT. These, and the tasks they involve, can most easily be understood in relation to a picture like the well-known “pyramid diagram” in Figure 1, probably first used by Vauquois (1968).

The simplest approach to translation is the so-called **direct** approach. Here the aim is to go directly from the source-language text to a target-language text essentially without assigning any linguistic structure. Since no structure is assigned, translation has to proceed on a word by word basis. Examples where this goes wrong are all too easy to find, and we will have little more to say about the approach. A more promising approach is base on the so-called **transfer** architecture. Here translation involves three main tasks:

- **Analysis**, where the source text is analysed to produce to an abstract representation or “interface structure” (IS) for the source-language text (IS_{SL}). This typically contains some properties of the source language (e.g. the source-language words).
- **Transfer**, where the source-language representation is mapped to a similar representation of the target-language text (IS_{TL}).
- **Synthesis**, or generation, where the target-language representation is mapped to a target text.

The third classical approach involves an **interlingual** architecture. Here the idea is

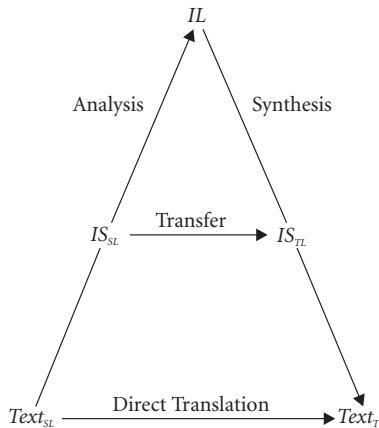


Figure 1. The “pyramid” diagram.

that one has at one’s disposal an “interlingua”: a more or less language-independent representation scheme. The role of the analysis component is to produce an interlingual representation (*IL*), which the synthesis component can map to a target language text.

A simple way to understand the relationship between these approaches is to start with the three tasks involved in the **transfer** approach, and say that the **interlingual** approach tries to eliminate the transfer task, and the **direct** approach tries to do without analysis and synthesis (i.e. it reduces everything to the transfer task).

This division into three tasks provides a rough classification of problems for what follows. In outline, the more “revealing and systematic” answer which was promised will be in four parts:

- Form under-determines content. That is, it is not always easy to work out the intended content from what is written. This is the *Analysis Problem* (Section 2).
- Content under-determines form. That is, it is difficult to work out how a particular content should be expressed (because there is more than one way to say the same thing in any language). We will call this the *Synthesis Problem* (Section 4).
- Languages differ. That is, that there are irreducible differences in the way the same content is expressed in different languages. We will call this the *Transfer Problem*, since in a transfer-based system it is typically where this problem shows up (Section 3).
- Building a translation system involves a huge amount of knowledge, which must be gathered, described, and represented in a usable form. We will call this the *Problem of Description* (Section 5).

Basing discussion around the tasks involved in the **transfer** architecture in this way may invite the question of whether one could not avoid the problems simply by eliminating corresponding the tasks. We will say something about this in relation to **interlingual** approaches in Section 3 (where we will argue that though they reduce the “**transfer problem**”, they do not eliminate it), and in Section 5, where we will look at recent “**analogical**” approaches, and argue that though they offer partial solutions to these problems, the problems themselves remain.

2. The analysis problem

The task of an analysis component is to take a source-language text (e.g. a sentence), and produce an abstract **representation** — the idea being that it will be easier to translate from this representation than from an unstructured string of source-language words. There will be different views on what sort of representation this should be (e.g. how abstract it should be), but it clearly must represent the “**content**” of the source text, since this is what the source text and its translation have in common.

The problem is to infer the content from the source text. There are two major difficulties:

- The source text will often contain sentences that are ill-formed, at least from the view point of the rules in an analysis component. Analysis components must be able to cope with this by being **robust**.
- The source text will often be **ambiguous**, so it may be difficult to work out what content is intended: the form of the input under-determines its content.

The problem of ambiguity is that no matter how superficial the representations we decide to use for an MT system, it will generally be the case that one string of words can correspond to several different representations.

The examples in (1) involve **lexical ambiguity**.

- (1) a. They are trying to design a better pen. ('writing implement' or 'animal enclosure'?)
- b. Our Jimmy has grown another foot. ('limb' or 'unit of measurement'?)
- c. The post has arrived. ('delivery of mail' or 'piece of wood'?)

The examples in (2) involve **structural ambiguity** — the indeterminacy of meaning is not due to any of the words, but to the different structures that can be assigned.

- (2) a. Concern has been expressed about conditions in the factory near the river that was polluted last week.

- b. The minister stated that the proposal was rejected yesterday.
- c. Sam has joined a student film society.
- d. Some young girls and boys have arrived.

Is it the river, or the factory that was polluted in (2a)? What occurred yesterday in (2b), the rejection, or the minister's statement? In (2c) is this a film society for students, or a society for student films (cf. *adult film society*)? Are the boys young, or is it just the girls in (2d)? The alternative interpretations of (2a) might be represented as (3).

- (3) a. the [factory near [the river] that was polluted last week].
- b. the [factory near [the river that was polluted last week]].

A very obvious and dramatic case of under-specification of content arises with pronouns, and other so-called anaphoric expressions. In an example like (4), one cannot tell who advocated violence: it might be the police, the women, or some other group that the speaker has mentioned earlier (or even a group that is being indicated in some other way).

- (4) The police refused to let the women demonstrate because they advocated violence.

A legitimate question in relation to these examples is: "Does it matter?" There are no doubt many languages where a straightforward translation would preserve the ambiguity of some or all of the examples above. In these cases, surely the ambiguity should just be ignored.

One difficulty with this is that the cases that can be dealt with in this "ambiguity preserving" way are not the same for all languages. Example (4) about the police refusing the women a permit because *they* advocated violence is unproblematic for translation into languages with only one third-person plural pronoun, but it is a problem in relation to languages like French which make a distinction according to gender (*ils* vs. *elles*).

A second difficulty is that the cases where one needs to worry are somewhat unpredictable even for one pair of languages. One might, for example, think the structural ambiguity in an example like (5) could be ignored, when translating into French (is Pauline writing in Paris, or are the friends in Paris?).

- (5) Pauline writes to her friends in Paris.

Indeed this ambiguity can be ignored with most verbs. But it must be resolved in translating a verb like *miss* (6), because its French translation, *manquer*, puts the noun phrase (NP) denoting the one who is missed in subject position, and realises the "miser" as a prepositional object. Thus, (6) has at least two non-equivalent translations, (7).

- (6) Pauline misses her friends in Paris.
- (7) a. *Les amis à Paris manquent à Pauline.*
the friends in Paris are-missing to Pauline
- b. *A Paris, les amis manquent à Pauline.*
in Paris, the friends are-missing to Pauline

There are two key points to note regarding the automatic treatment of ambiguity. The first is that ambiguities combine to produce a combinatorial explosion. Consider a ten-word sentence, where each word is two ways ambiguous. Even before we consider structural ambiguities, or those with some other source, we have 2^{10} possibilities. Suppose there is a verb, followed by an NP, and a prepositional phrase (PP) (like example (6)). This gives an additional ambiguity, because the PP can be part of the NP, or not. So we may have as many as $2^{10} \times 2$ possibilities. If there is another PP the possibilities increase further. Now consider the pronoun *her*. It could, potentially be referring back to any female individual mentioned earlier. In the worst case, all these sources of ambiguity would be independent, and one is faced with a combinatorial explosion.

Fortunately, the worst case does not always arise because some of the ambiguities cancel out. In isolation either *loves* or *presents* can be a verb or a noun, but in (8), *loves* must be a verb, and hence *presents* must be a noun.

- (8) Sam loves presents.

Nevertheless, in practice the number of possibilities is still very large, partly because most sentences are much more than ten words long, and most words are more than two ways ambiguous. It is reasonable to expect tens or even hundreds of analyses for quite ordinary sentences.

The second key point relates to the variety of information that would be required to disambiguate examples like these. Example (9) is very similar to (2a), but it is *unambiguous* because of grammatical information (the presence of a plural verb *were* unambiguously picks out the plural *factories* as the grammatical head of its subject, so the interpretation is *factories ... were polluted*). Thus, grammatical/structural information has a role to play.

- (9) Concern has been expressed about conditions in the factories near the river that were polluted last week.

Similarly, (10) is unlikely to be interpreted as ambiguous, because of common-sense knowledge about the relative sizes of sheep and writing pens vs. animal pens (and the fact that putting A inside B entails A being smaller than B):

- (10) Sam put the sheep in the pen.

Likewise, *young girls and boys* is ambiguous, but *pregnant women and men* is not, because as a matter of fact, men do not become pregnant, and the reading one prefers for (4) will depend on all sorts of assumptions about women and police, and what constitute grounds for refusing permission to demonstrate.

In principle, it seems that any grammar fact or fact about the world, any piece of information or common-sense inference could be required for the correct resolution of an ambiguity.

Turning to the problem of **ill-formed input**, it is an unfortunate fact that ordinary written language, even the kind that has been carefully edited and prepared (like the contents of this book) abounds in errors of spelling, repeated words, transposed words, missing words, and what will appear to an analysis component to be errors of grammar.

Solutions (at least partial solutions) to these problems are not hard to find. For example, if we fail to produce an analysis for a whole phrase or sentence, we may nevertheless have successfully analysed parts of it, so we might try to hypothesize a missing word, or transpose a pair of words, and try to re-analyse, using the partial analyses that have been established. In a case like (11), we might just relax the requirement that a third-person singular subject requires a particular verb form. Of course, such tricks are a long way from the flexibility of the human reader, which is based on an overall understanding of the text.

- (11) The problems are interesting, but the solution (*sic*) leave something to be desired.

However, two points should be kept in mind. First, inserting words, trying permutations of words and so on, are all potentially combinatorially explosive. Second, notice how dealing with ill-formed text interacts with the problem of ambiguity. The obvious way to deal with a case such as (11) is to disregard the rules that enforce subject–verb agreement. But doing this generally will lead to increased ambiguity. In particular, the unambiguous example (9) becomes ambiguous if one ignores the information about subject–verb agreement, because it becomes indistinguishable from (2a). In principle, this point holds for any restriction at all: imposing the restriction may lead to a failure in analysis; relaxing it will lead to more ambiguity.

All together, the problems posed by ambiguity and robustness may make the situation look rather desperate: reliable analysis seems to require nothing less than complete understanding at a level comparable to a human. Indeed, such considerations led some early researchers to declare that MT was not just difficult, but theoretically impossible. Fortunately, things are not quite this bad in practice. Partly, this is because, as noted, some ambiguities “cancel out”, and some can be

excluded by employing perfectly normal grammar rules (subjects and verbs agree in person and number). Restricting the domain and or text-type that one deals with will also be helpful. Some of the problems can be addressed by clever interaction with a human operator or post-editor (e.g. pronouns can be left flagged and left untranslated for a person to fix). If all else fails, one can just choose one interpretation, either at random, or on the basis of some ideas about which interpretations are more likely than others — this will be wrong some of the time, but most of the time it will be right.

3. The transfer problem

The task of a transfer component is to take the sort of abstract representation produced by the source-language analysis component (call this a “source IS”), and produce something that can be input to the synthesis component of the target language (call this a “target IS”). Obviously, the closer the two ISs, the easier this will be. The transfer problem is that they cannot be the same, because languages do not associate form and content in the same ways. Thus, rules must be written to relate source and target ISs.

To be concrete, let us assume that ISs are relatively superficial representations, along the lines shown in (12) and its translation (13).

- (12) a. I miss London.
b. [_{sentence/pres} miss,
 [_{np/sing/1st} PRO],
 [_{np} London]]
- (13) a. *Londres me manque.*
b. [_{sentence/pres} manquer,
 [_{np} Londres],
 [_{np/sing/1st} PRO]]

Given this sort of representation, the sort of thing transfer rules need to say is that *London* translates as *Londres*, that first-person singular NPs translate as first-person singular NPs (usually), and that translating a structure where the verb is *miss* involves getting the translation of its subject (the first NP) and its object (the second NP), and putting them in the appropriate slots in a structure whose verb is *manquer*, namely the indirect-object and subject slots respectively, and dealing with the tense, and so on.

The assumption is that though languages use different words and structures to express the same content, nevertheless there are enough similarities that words and

structures can be put in some kind of fairly straightforward correspondence. Of course, this can easily lead to “translationese” where the structure of the source language is wrongly carried over to the target language. Nevertheless, the assumption holds and can be the basis of reasonable translation for many constructions and languages. Unfortunately, there are also many cases where the assumption fails.

Sometimes languages either package content differently, or just use radically different structures to express the same content. A case of the former can be seen in (14), which exemplifies a general difference between the way information about the direction and manner of motion is packaged in English and French (and other Romance languages). In English, the manner and motion are expressed in one item (the verb *run*), the direction is expressed by a PP (*into the room*). In French, the verb *entrer* ‘enter’ expresses motion and direction, while manner is expressed by an adverbial (*en courant* ‘by running’).

- (14) a. He ran into the room.
- b. *Il entra dans la chambre en courant.*
 he entered into the room by running

Moreover, while it is possible to write a (rather complex) transfer rule that will state the correspondence here, this is in fact a quite general phenomenon, and it would be nice to have a general treatment, rather than dealing with individual cases (one rule for *run into*, one for *walk into*, one for *fly out of*, etc.)

A case of languages using radically different structures for roughly the same content can be seen in (15). Dutch (15a) involves a construction with an impersonal pronoun, Spanish (15b) uses a reflexive, and English (15c) uses a passive construction. If the corresponding IS representations are as superficial as those above, some very complex transfer rules will be required.

- (15) a. *Man verkoopt hier appels.*
 one sells here apples
- b. *Se venden manzanas aquí.*
 self they-sell apples here
 Lit. ‘Apples sell themselves here’
- c. Apples are sold here.

The need for very complex rules can also arise when two languages have corresponding constructions (i.e. content is packaged similarly), but the constructions are subject to different grammatical restrictions.

One example of this involves adjectives like *difficult* and *easy* and their translations in German. In (16a) the subject, *Sam*, is understood as one of the objects of the verb *convince*: compare (16b). The German (16c) is structurally parallel, and expresses the same content.

- (16) a. Sam is easy to convince.
b. It is easy to convince Sam.
c. *Sam ist einfach zu überzeugen.*

Unfortunately, there are differences between this construction in English and German. One difference is that while in English the understood position can be any kind of object, in German it must be a *direct* object. Thus, a straightforward translation of (17) produces the ungrammatical (18a). Instead, one must produce something like (18b), with a very different structure.

- (17) Sam is easy to work with.
(18) a. **Sam ist einfach mitzuarbeiten.*
b. *Es ist einfach mit Sam zu arbeiten.*
It is easy with Sam to work

It is important to notice that even apparently small differences between languages can give rise to problems. In English, the idea of being hungry is expressed with an adjective, in German a noun is used, as in (19).

- (19) a. I am hungry.
b. *Ich habe Hunger.*
I have hunger

Not much to worry about here, one might think: a rule to the effect that English *hungry_A* translates as German *Hunger_N* should be sufficient. Sadly, this is not the case, as one can see from an example like (20) where the English adjective is “intensified” with a word like *very*. One cannot simply get the normal translation of *very* (*sehr*): instead the adjective intensifier *very* must be translated as a nominal intensifier *viel* ‘much’.

- (20) a. I am very hungry.
b. *Ich habe viel Hunger.*
I have much hunger

Often these “knock-on” effects of the way content is expressed require information that is absent in one language to be supplied in another. A simple case of this arises in the translation of German examples like (21a) into English.

- (21) a. *das für Sam neue Auto*
the for Sam new car
b. the car which is new to Sam

The problem is that English does not allow nouns to be pre-modified in the same way as German (cf. * *the new to him car*). The solution is to make the modifying

material into a post-modifier, putting it after the noun. This sounds easy enough, but moving the material after the noun involves turning it from a PP into a relative clause, and turning it into a relative clause involves supplying a verb (*be*), and when one supplies a verb one is also required to supply a tense (in (21b) we assumed it was present tense, but there is nothing in the German to indicate this).

A sort of limiting case of differences between constructions arises where one language completely lacks a construction, and one must always resort to finding a paraphrase. French simply lacks a resultative construction corresponding to (22a), so a straightforward translation is impossible. Instead of a simple adjective (*flat*), a whole subordinate clause is required, for which a tense, and a subject must be supplied (22c).

- (22) a. They hammered the metal flat.
 b. **Ils ont martelé le métal plat.*
 c. *Ils ont martelé le métal jusqu'à ce qu'il est devenu plat.*

'They hammered the metal until it became flat'.

Of course, the need to supply information that is unspecified in the source structure does not arise just because of particular constructions. It can arise between languages generally. For example, in English, one cannot avoid the issue of whether an NP is singular or plural, and whether it is definite or indefinite. In Japanese, on the other hand, this information can remain unspecified, so there is a clear problem in translating from Japanese into English. There is a similar problem going from English to Japanese, because in Japanese it is hard to avoid being precise about social relations between the writer and reader (e.g. it affects the form of the verb) which are not expressed in the English.²

It is perhaps easy to see the general direction of a solution to these problems. The transfer problem arises because source and target language interface structures (ISs) differ. The more similar they are, the smaller the problem should be. Does this mean that one can avoid the problem entirely by adopting an interlingual approach? Unfortunately, it does not. The reason is that even under an interlingual approach it will be very difficult to ensure *identity* of source and target ISs.

First, however, it is worth noting a drawback to an interlingual approach, namely that making source and target representations more similar complicates analysis, by increasing ambiguity, sometimes unnecessarily. A simple example arises with languages (like Japanese and Chinese) which have different words for older and younger sister. This distinction will have to be present in an adequate interlingual representation for such languages. This means that producing an interlingual representation for English *sister* will involve disambiguation (older or younger sister?). This is entirely appropriate when the target language is Japanese or Chinese, but it is wasted effort when the target is another European language. (As

will become clear in the following section, adopting more abstract representations also complicates synthesis).

The following example will clarify the difficulty of ensuring *identity* of source and target representations. An approximate rendering of the content of English (23a) might be as in (23b), which says that for all events e , if e is an eating event where the thing doing the eating is Sam, then the eaten object (f) is fish.

- (23) a. Sam eats only fish.
b. Forall e : if [eating(e) & eater(e ,sam)]
then [eaten-object(e,f) & fish(f)]

The same idea is expressed in Japanese as (24a), whose content is most naturally given as something like (24b), which says that “there are no eating events with Sam as eater that do not involve fish as object” (one reason for regarding this as a “natural” representation is that it correctly captures the negative nature of the Japanese sentence).

- (24) a. *Sam wa sakana shika tabenai.*
Sam topic fish apart eat-not
‘Sam does not eat anything apart from fish.’
b. Not Exists e : [eating(e) & eater(e ,Sam)]
& not [eaten-object(f) & fish(f)]

Now these representations are equivalent. However, they are not identical, and it would clearly be difficult to find a general way of ensuring that this sort of thing does not arise. Not only would representations have to be very abstract, they would look utterly arbitrary from the point of view of some languages. (Why should *Sam eats only fish* involve a negative? Why should the Japanese *not* involve a negative?)

However, given the equivalence of these ISs, one might still hope to do away with transfer rules by formulating a general “inference” procedure along the following lines: take the source IS, input it directly to the synthesis component, if a correct target sentence is produced, then stop. Otherwise, find an equivalent IS, and try with that, etc. There are two worries here. First, it assumes we have a “logic” for ISs, which provides a well-defined notion of equivalence for ISs. Second, finding an equivalent IS is very likely to be one of the problems for which solutions cannot always be computed (because the number of equivalent ISs is likely to be infinite). It is, in any case, a combinatorially explosive process.

Thus, while using more abstract representations is clearly a good idea, because it will make transfer rules simpler, and while the transfer problem can be simplified by the right choice of representations, the implication of this argument is that there are irreducible differences in the way languages express “the same” content, and the transfer problem cannot be completely eliminated.

4. The synthesis problem

The two aspects of the synthesis problem are actually instances of the last problem discussed in the previous section. There are typically many ways in which the same content can be expressed. In short: meaning under-determines form.

The first aspect of the problem is that sometimes only one of the ways of expressing the content is correct. There seems to be no principled reason why (25a) is correct in English, rather than (25b,c).

- (25) a. What time is it?
- b. How late is it?
- c. What is the hour?

On the face of it, these would be equally good ways of expressing the same content. It is just that only one is idiomatic English. The solution to this problem may look simple — just keep a list of the contents that must be realized by these semi-fixed expressions, and stop rules applying to produce the correct, but unidiomatic alternatives. But this solution is not foolproof, precisely for the reasons discussed at the end of the previous section: there are many ways in which the content that one would like to realize as (25a) could turn up in an IS representation, so it will be hard to list them all.

The second aspect of the synthesis problem is in some ways the converse of the first. It occurs when there is *no* obvious way of selecting the right way to express the content. To take a very simple example, the content of (26a) might be represented as (26b),

- (26) a. Sam saw a black cat.
- b. Some e : seeing(e), by(e ,Sam), of(e , y), cat(y), black(y), before(e ,now)

i.e. there is a seeing event (e), where Sam did the seeing, and the seen thing (y) was a black cat, and the event occurred before now.

This content can be expressed in English in many other ways (27).

- (27) a. Sam saw a cat. It was black.
- b. Sam saw something black. It was a cat.
- c. Sam saw a cat which was black.
- d. Sam saw a black thing which was a cat.
- e. A black cat was seen by Sam.
- f. Something happened in the past. Sam saw a cat.
- g. There was a black cat. Sam saw it.
- etc.

The problem is how to select among these alternatives. In part, this is just another combinatorial problem: there are just too many alternatives to consider. But more serious is the problem that it is hard to know in general when one way of saying something is better than another. The only reliable test is to read what has been produced, and see if it is clear, and would be clear to a potential reader. But this is certainly asking too much of a computer. We would be asking not only that it understand sentences, but also that it should be able to consider whether someone else would be able to understand them.

Of course, one approach to this problem is to say “choose the output that is most similar to the source text.” This is, in fact, one of the ideas behind a transfer-based approach using fairly superficial structures: by staying close to the surface, surface information from the source language is preserved, and the synthesis problem is made easier. But this will also lead to there being more differences between source and target language structure (cf. the transfer problem).

5. The problem of description

The discussion so far has noted a number of fundamental, more or less theoretical, problems. The purpose of this section is to point out that even if satisfactory (or anyway workable) solutions to these problems can be found, building MT systems would still be a difficult task, because of the difficulty of gathering and describing the knowledge required for translation, in the form of sufficiently explicit rules.

One aspect of the problem here relates to the number of languages one may have to deal with. The analysis–transfer–synthesis approach requires an analysis and synthesis component for each language, and a transfer component for each pair of languages. For n languages, there are $n \times (n-1)$ such pairs (not n^2 , because we do not need a transfer component from any language into itself). Of course, one may expect that a lot of the transfer rules taking English to French may be workable in reverse. So one may be able to divide this number by 2. Nine languages still need 36 transfer components, 20 languages need 190 transfer components.

Moreover, these transfer components will tend to be large. At the very least, one can expect there to be rules that say how individual words are translated (*girl* translates as *fille*, *house* as *maison*, etc.), so there will be at least as many transfer rules as there are source-language words to be dealt with. Unless we are dealing with a very small domain (e.g. weather reports), this is likely to be in the tens of thousands.

Of course, one can try to eliminate the transfer components (e.g. adopting an interlingual approach). But one of the lessons of the preceding sections is that even if this is possible it will complicate analysis and synthesis. While it seems certain that

shifting work from transfer into analysis and synthesis must make sense in the long run, it is an open question at what point the advantages will appear in practice. (For three or four languages or where languages are very similar, like Italian and Spanish, or Dutch and German, there may be little advantage; on the other hand, trying to write transfer components for even a few tens of languages would be a huge undertaking.)

In any case, this still leaves the aspect of the problem that relates to the size and complexity of the rules required for each language, i.e. for analysis and synthesis. Again, one might hope to find some similarities, but one cannot expect analysis and synthesis rules for one language to be identical.³ In any case, one needs at least: a lexicon, a set of **morphological rules**, and a set of **syntactic/semantic rules** (one might also need rules describing discourse structure, or document formatting, as well as other things).

The lexicon contains a description of all the basic words the system is to deal with (their grammatical category, spelling, what they correspond to in the abstract representation), what complements they take (e.g. whether they are transitive or intransitive), any idiosyncrasies of syntax or morphology. The morphological rules describe the ways in which different forms of words are formed (e.g. plural formation: *boy* → *boys*, *bus* → *buses*, *child* → *children*) and the ways in which new words can be formed, e.g. by compounding (combining two independent words like *film* and *society* to make *film society*) or affixation (adding *-ize* to *legal* to make *legalize*, and then adding *-ation* to make *legalization*). The syntactic/semantic rules describe the way in which words and phrases can be combined together to make larger phrases. Of course, in each case, the rules have to specify not only what can be combined with what, but what sort of abstract representation should be built.

In a reasonably sized system, one will certainly be dealing with tens of thousands of words, and with several hundred morphological and syntactic rules. Even leaving aside the fact that writing some of these rules requires fundamental research (e.g. the only morphological description that exists for a language may be at the level of a pedagogical grammar, which is a huge distance from the level of explicitness needed for computational implementation), one is clearly looking at tens of person years of effort by highly trained linguists for each language just to describe the requisite linguistic knowledge.

There are three ways of trying to minimize this problem.

1. Restrict the coverage of MT systems to very specialized domains, where vocabulary is small and the grammar is relatively simple.
2. Exploit existing sources of knowledge, for example automatically converting machine-readable versions of monolingual or bilingual dictionaries for use in MT systems.

3. Try to manage without explicit representations of linguistic (or non-linguistic) knowledge at all.

The first solution is attractive in theory, and has proved successful in practice (cf. the outstanding success of *Météo* — see Chapter 15), but its value is limited by the number of such domains that exist (it has proved very difficult to think of other domains that are as tractable as weather reports). The problem with the second solution is that existing dictionaries and grammars have normally been created with *human* users in mind, and so do not contain the kind or level of information required for use in MT. The third solution underlies one of the recent approaches which are discussed in the following section.

6. Other approaches

The preceding sections have looked at the problem of MT in terms of the “classical” approach, where translation takes place in three (or possibly two) stages, involving representations and explicit rules encoding various kinds of linguistic and other knowledge. The last decade has seen the emergence of so-called **analogical** approaches to MT, which, at least in their radical form, dispense with the representations and rules. The possibility arises that such approaches thereby solve some or all of the problems. This section will show why this is not the case, or at least why it is only partly the case. The analogical approaches in question are **example-based** approaches and stochastic or **statistical** approaches.

6.1 Example-based MT

The leading idea behind so-called Example-based MT (EBMT) approaches is that instead of being based on rules, translation should be based on a database of **examples**, that is, pairings of fragments of source- and target-language text (see also Chapter 3.6). Suppose, for example, that one has the pairings in (28) and (29) in the database, and has to translate (30) from English into French.

- (28) I have a headache.

J'ai mal de tête.

I have ache of head

- (29) I'd like something for a hangover.

Je voudrais quelque chose contre la gueule de bois.

I would-like some thing against the face of wood

- (30) I have a hangover.

Ideally, what should happen is that matching (30) against the English parts of (28) and (29) will reveal that *I have* can translate as *J'ai*, and *a hangover* can translate as *la gueule de bois*, which can be combined together to produce the acceptable translation (31).

- (31) *J'ai la gueule de bois.*

This is a very intuitive and appealing model of translation (for example, it seems to reflect something of the way humans work). In the best case, we may get an exact match for an input sentence, which will be paired with just one translation, and all the problems are solved. Unfortunately, in the general case things will not be so simple, and all the problems remain.

First, even when we have exactly matched an input sentence, it may correspond to several target examples, among which we must choose (if the database is sufficiently large and representative, genuinely ambiguous examples will get alternative translations). If the alternatives are equivalent, we have an instance of the synthesis problem of Section 4; if they are not equivalent, we have an ambiguity problem, analogous to the analysis problem of Section 2 (the difference is that we have to chose between alternative examples, rather than alternative representations). “Ambiguity” will arise in other ways, because there will typically be many other examples that partially match an input like (30), for example, those in (32) and (33). Each of these will suggest an alternative translation for *I have*, which will not yield correct translations (e.g. *Je suis* ‘I am’ and *Je viens* ‘I come’).

- (32) I have left.
Je suis parti.
 I am departed
 (33) I have just left.
Je viens de partir.
 I come of to-depart

Moreover, given that (28) has been chosen as a match for part of the input (30), we have to decide which parts of the French translation to take: how do we decide that *J'ai* corresponds to *I have*? This is like the transfer problem of Section 3 in that it will be harder to work out correspondences the more source and target examples diverge from word-for-word alignment (i.e. the more the languages diverge in the way they express content). Finally, having decided which pieces of the French examples we need to combine, how do we decide to combine them as in (31), rather than in the other order, or somehow mixed up? This is again somewhat analogous to the synthesis problem of Section 4.

In principle, one might still hope to manage these problems without recourse to rules (i.e. explicit linguistic knowledge). For example, one might observe that the

sequence *ai mal* ‘have ill(ness)’ occurs much more frequently than *?viens de mal* ‘comes of ill(ness)’, and on this basis choose the correct (31) over the incorrect (and meaningless) (34).

- (34) **Je viens de la gueule de bois.*

However, this leads us directly to the EBMT version of the problem of description, which is that in order to make the approach work one will need many millions of examples. Bilingual dictionaries provide one source of appropriate examples, but this will yield at most a few thousand. For the rest, we must rely on “aligned corpora”, of which we will say more shortly.

6.2 Statistical approaches

The intuitive appeal of statistical approaches can be seen when one considers how one normally approaches very complex processes involving a large number of interacting factors. One approach is to try to disentangle the various factors, describe them individually, and model their interaction. One might, for example, try to model the way a crowd behaves by trying to understand how every individual in it behaves, and how they interact. But for many purposes, including the case of crowd behaviour, it is more sensible to step back and try to model all or part of the process statistically. Given that translation is a very complex process involving many factors, the appeal of some kind of statistical methodology should be clear.

Of course, there are many ways one could try to apply statistical methods in a “classical” approach to MT, but a more radical idea has also been proposed. The central idea is this. When presented with a French sentence *f*, we imagine that the original writer actually had in mind an English sentence *e*, but that *e* was somehow garbled in translation so that it came out as *f*. The job of the MT system is just to produce *e* when presented with *f*. Seen in this way, translation is an instance of transmission down a **noisy channel** (like a telephone line), and there is a standard technique that can be used to recover the original input (the English sentence *e*), at least most of the time. The idea is that *f* is more or less likely to occur depending on which English sentence the writer had in mind. Clearly, we want the one(s) that give *f* the highest **probability**. Moreover, it also makes sense to take into account the relative probabilities among the English sentences (perhaps the probability of getting (35a) given (35b) is not much different from that given (35c) but the former has a higher probability, and is the right choice of course).

- (35) a. *Quelle heure est-il?*
b. What time is it?
c. What hour is it?

To make this work, we need: (a) a statistical translation model, which assigns a probability to f for each sentence of English; (b) a monolingual statistical model of English, which assigns a probability to every sentence of English; and (c) a method for finding the best candidate for e according to these models.

Notice that since in principle *any* English sentence could give rise to f , (c) is a combinatorially explosive problem *par excellence* (even if we restrict ourselves to sentences that are about the same length as f , there will be millions of possibilities), but if we can find a (presumably imperfect) way of searching through the vast number of possibilities, we have a method that works without rules (hence no problem of description), without analysis (no problem of robustness — even completely ungrammatical sentences have *some* probability, however small), without intermediate representations (hence no problem of ambiguity deciding which representation to assign), and no problem of synthesis (deciding which sentence to produce given a particular representation).

Sadly, there are two important reasons why this is not a panacea. The first relates to a different version of the problem of description. The second relates to the quality of the available statistical models.

The statistical version of the problem of description is the problem of **sparse data**. Consider just the model of English: the only way that we can be sure that, say, (35b) is more probable than (35c) is by analysing huge amounts of text, and seeing that *time* appears more often in this context than *hour*. The problem is that in order to do this for most expressions we will need to examine astronomically large amounts of text. Even if one looks at many millions of words, many words appear only once or twice. So, there is a real problem getting reliable statistics. The problem is worse still when one considers translation. Here one relies on **aligned parallel corpora**, that is, collections of texts in two languages which are supposed to be translations of each other, which have been aligned so that it is simple to find the target-language sentence(s) that translate any particular source-language sentence. The classic example is the Canadian Hansard Corpus, consisting of reports of proceedings in the Canadian Parliament, which are published in both English and French. But such corpora are rare (non-existent for many pairs of languages), and tend to be relatively small. And of course the translation model typically needs more data than the monolingual model (whatever the probability of seeing an expression on its own, the probability of seeing it as the translation of some other expression must generally be lower).

A standard example of a monolingual statistical model is a so-called **bigram** model like those which have been very successfully applied in speech recognition. They involve the simplifying assumption that the probability of any given word sequence can be identified with the joint probability of each word occurring, given that the preceding word occurred. The probability associated with *The cat died* is

the joint probability of *The* occurring as the first word in a sentence, of *cat* occurring given that the preceding word was *The*, and the probability of *died* occurring given that the preceding word was *cat*. The basic data for such a model is thus observations about the relative frequency of various pairs of words (bigrams). A generalization of this for the translation case might assume that the probability of *f* appearing as the translation of word *e* depends on the predecessors of *f* and *e*. But of course, it is clear what is wrong with this model. While the probability of *cat* clearly is influenced by the probability of *The*, this is not because *The* is the word before *cat*, but because *cat* is a noun, and nouns appear in NPs, which often start with determiners (like *The*). For example, *The* is exerting the same sort of effect in an expression like *The big cat* or *The big fat black cat*, while in a bigram model the effect falls off dramatically (it is seen as depending on the likelihood of *big* following *The*, and *fat* following *big*, and *cat* following *fat*). Of course, we can replace the bigram model with one that takes account of the grammatical structure, but now we are back with at least some of the ambiguity problems again, because taking account of the grammatical structure will involve linguistic rules and representations. Moreover, the statistical version of the description problem will be worse, because we need statistics not just about what words come next to each other, but about the what structures go with what strings of words. Such statistics will be hard to find, because they will require text that has been analysed and given a representation (and giving text the *right* representation takes us straight back to the problem of ambiguity again).

Notice that the value of statistical methods *per se* is not at issue here, because to use statistical methods it is not necessary to adopt such a radical stance. One might, for example, try to use such methods to achieve robustness and disambiguation in analysis (e.g. if one encounters a lot of finance-related words in a text, it is quite likely that an occurrence of *bank* will denote a financial institution). But they are not a panacea, because a statistical method is only as good as the statistics, and these depend on what factors the model takes into account, and on the amount and quality of the data. Once one accepts the need for abstract representations, one is immediately and inevitably faced with the problems discussed in the rest of this chapter. Statistical methods are a contribution to the solution, but they are not in themselves the solution one might have hoped.

7. Conclusion

In short: translation is about producing a target text that preserves the content of the source text. This is difficult for computers because (a) form under-determines content; (b) content under-determines form; (c) languages differ in the way they

express content; (d) it is difficult either to express the principles involved with the necessary precision, or to find the data needed for a statistical approximation.

Further reading

The reader who wants more background, details, or other perspectives can find extended, but still introductory, discussion of the issues discussed here, as well as references to the primary literature in, *inter alia* Arnold et al. (1993), Hutchins and Somers (1992), Kay et al. (1994) and Whitelock and Kilby (1995), and elsewhere in this volume, of course. More advanced and technical discussion can be found in Trujillo (1999).

Notes

1. This is much simpler than the task we started out with, which is one reason that MT is not a serious threat to the employment prospects of human translators.
2. In fact, this is not a new problem: it is just an instance of the ambiguity problem discussed above. The difference is that questions such as whether writer and reader are being polite or familiar only appear to be ambiguities when one thinks about translation into a certain language. But as with resolving ambiguity in analysis, inferring the necessary information can require nothing less than full understanding of the text (and context).
3. For example, one may want analysis to accept ungrammatical inputs, which one does not want synthesis to produce.

References

- Arnold, D., L. Balkan, R. Lee Humphreys, S. Meijer and L. Sadler (1994) *Machine Translation: An Introductory Guide*. Manchester: NCC Blackwell.
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*. London: Academic Press.
- Kay, Martin, Jean Mark Gawron and Peter Norvig (1994) *Verbmobil: A Translation System for Face-to-Face Dialog* (CSLI Lecture Notes No. 33). Stanford, CA: Center for the Study of Language and Information.
- Trujillo, Arturo (1999) *Translation Engines: Techniques for Machine Translation*. London: Springer Verlag.
- Vauquois, Bernard (1968) "A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation", *IFIP Congress-68*, Edinburgh, pp. 254–260; reprinted in Ch. Boitet (ed.) *Bernard Vauquois et la TAO: Vingt-cinq ans de*

- traduction automatique — analects*, Grenoble (1988): Association Champollion, pp. 201–213.
- Whitelock, Peter and Kieran Kilby (1995) *Linguistic and Computational Techniques in Machine Translation System Design* (2nd edition). London: UCL Press.

CHAPTER 9

The relevance of linguistics for machine translation

Paul Bennett

UMIST, Manchester, England

1. Introduction

In this chapter we consider the various ways in which linguistics, the scientific study of language, can be exploited in MT systems. Linguistics can be used by MT system developers in a rather random way, as a source of analyses or ideas for solving a specific problem. The earliest, “direct” systems may be said to have at best taken this kind of approach, and at worst to have ignored linguistic findings altogether. In this chapter, however, we shall be concerned with more rigorous and systematic uses of linguistics.

Linguistics is concerned with providing descriptions of languages, theories of human language in general, and **formalisms** within which these descriptions and theories can be stated. Linguistics is relevant to *human* translation as well as MT, though we will be covering mainly the latter here.

This chapter is structured as follows. In Section 2 we present an overview of approaches to linguistics and of how MT can make use of this discipline. In Section 3 we examine ways in which linguistics can contribute to defining the abstract representations used in MT, and in Section 4 take this further by looking at examples where source and target sentences differ considerably in structure. Section 5 studies the translation of tense, and Section 6 deals with functional aspects of language and the extent to which MT systems can incorporate these. Section 7 is a brief conclusion.

2. The field of linguistics

This section examines some of the ways in which linguistics can feed into MT research. To assist the reader, we can summarise the main contrasts we shall set up as follows:

- new framework vs. use of existing framework
- formal vs. functional approaches
- representations vs. descriptive components
- monolingual vs. multilingual or contrastive.

Two approaches to systematic exploitation of linguistics may be distinguished. In the first, a new linguistic **framework** is established, designed specifically to cope with the translation problem, and drawing insights from a variety of linguistic theories. Such an eclectic approach implies a great deal of foundational research to create a usable framework, and descriptive work to write the components of the system. In the second, an existing linguistic **theory** is taken over more or less wholesale, though usually supplemented by extra mechanisms for handling transfer or choosing the most likely interpretation of ambiguous sentences. This has the advantage that not just the theoretical foundations but also previously written grammars and dictionaries can be employed, preferably with little need for adaptation. It thus satisfies the criterion of **reusability**, an important concept in current computational linguistics, whereby resources created for one purpose can be used again in another system or application. The first of these two approaches has, however, been by far the most widely used, if only because linguistic theories of sufficient breadth and robustness have only been developed in recent years. Accordingly, here we shall concentrate on the first approach, though with some passing references to the second, which is becoming more popular.

At least two main “schools” can be distinguished in linguistics. In **formal** approaches, the emphasis is on explicit description of the structure and meaning of words and sentences. Noam Chomsky’s theory of “generative grammar” is the best-known representative of this school. In contrast, **functional** approaches are more concerned with the use of language and the ways in which sentences are combined together to produce a well-formed text. Formal frameworks are far more easily incorporated into software (more **computationally tractable**) than functional ones, and have been more influential in MT research and development. We shall therefore focus on them here, but again will refer to functional ideas as well, especially in Section 6.

Most linguistic frameworks draw a fundamental distinction between the **grammar**, a set of rules or principles capturing general facts about the language, and the **lexicon** or dictionary, a list of idiosyncratic facts about words. Theories draw the line between grammar and lexicon in different places, and it is generally claimed nowadays that generalisations can be stated in the lexicon as well as in the grammar. Nevertheless, it is convenient to draw some such distinction, and it is a standard one in MT and in computational linguistics in general, so we shall assume it here.

Within the grammar, three different **levels** of description can be distinguished.

Morphology is concerned with word structure, syntax with sentence structure, and semantics with meaning. It is syntax and semantics that form the core of MT systems and will therefore be the focus of attention in what follows, but morphology also raises a number of translation problems, as seen, for instance, in novel words like *transferee*, *Murdochization* and *dome fiasco*.

There are also two fundamental ways in which linguistics can feed into MT research. The first is by way of **representations**. Whether in a transfer-based or interlingua system, a sentence is converted to some representation of its structure or meaning. Linguistic notions can play a crucial role in determining what such representations look like and what representation is appropriate for a particular example. The other is in terms of **description**, i.e. the modules of the system which describe or capture various kinds of knowledge — the grammars, lexicons or transfer components. The various entries in a lexicon, for instance, may be structured along lines specified by some linguistic theory. Since the details of lexicons and grammatical rules are inevitably more specific to individual systems, we concentrate here on representations (see Sections 3 and 4).

Linguistic research and description can also be concerned primarily with a single language, or can be multilingual. Monolingual work inputs to MT in being relevant to analysis and synthesis (or generation), but naturally it is multilingual linguistics which is of most pertinence. For example, work in **language typology** studies ways in which languages differ from each other as well as what all languages have in common. However, much work within this paradigm, which seeks to derive statements of the form, “If a language has property X, then it is certain or highly likely to have property Y as well”, is not as useful for MT as it might be, since it does not provide the explicitly contrastive kind of knowledge that is crucial. The language-teaching tradition of **contrastive analysis**, which is specifically bilingual, has been relatively under-utilised in MT, but may have more to offer. After all, making generalisations that help a learner avoid mistakes in a second language is not all that different from writing rules that enable an MT system to produce adequate translations. Indeed, James (1980: 4) sees contrastive analysis and translation theory as two of the branches of the common endeavour of “interlingual” linguistics.

To illustrate, let us consider an example from James (1980: 67–70). As seen in (1), in Portuguese (and in a number of other languages), predicate nominals — those following the verb *be* — have no article, except when modified by an adjective.¹ He states the transfer rule in (2) for English to Portuguese. This rule states that the English indefinite article *a/an* has no equivalent (“Ø”) in Portuguese when the noun is not preceded by the adjective: the slash “/” shows the context, the dash “—” indicates the position of the article, the box marks the crucial feature in the rule, and the minus “–” denotes absence of a type of word, in this case an adjective.

- (1) a. *Ele é professor.*
he is teacher
'He is a teacher.'
b. *Ele é um bom professor.*
he is a good teacher
'He is a good teacher.'
- (2) Indefinite article → Ø / — [— Adj] N

This rule needs to be made more precise by referring to predicate nominals only, but its similarity to the kind of statement needed for MT should be clear. However, it is stated purely as a relation between surface strings and does not take advantage of the possibility of using more abstract representations (see next section).

3. Abstract representations

One major way in which linguistics is utilized in MT relates to the representations found in transfer and interlingua systems (see Chapter 8). Both rely on some abstract representation of sentences as the result of analysis and the input to synthesis or generation. The transfer and interlingua architectures place rather different demands on these representations, but it still seems fruitful to discuss the properties of these abstract **interface structures** in terms general enough to cover both possibilities (though with a bias towards transfer).

The general idea is that a sentence is stripped down to its bare bones, i.e. the lexical-class words (nouns, verbs, adjectives, adverbs), which basically describe entities, actions and their properties. Grammatical words (articles, conjunctions, some prepositions) are converted to **features** attached to lexical words, as are many affixes. An example will make this clearer. Sentence (3a) essentially consists of the action and entities picked out in (3b).

- (3) a. A plumber mended the sink.
b. plumber – mend – sink

A possible abstract representation is given in (4).

- (4) predicate: *mend* (past)
subject: *plumber* (singular, indefinite)
object: *sink* (singular, definite)

Here we have indicated the grammatical **function** of the expressions (predicate, subject or object), making word-order less crucial, and have shown the articles and the morphological properties of the nouns as features (extra specifications, given in

brackets). Converting lexical words to a **citation form** makes lexical transfer much easier, and is essential in languages where there are large numbers of inflectional forms (cf. the dozens of forms for verbs in Italian, depending on person, number, tense and mood). Computational techniques for this process — sometimes known as **lemmatization** — are now well-advanced.

More controversially, we have also ignored **syntactic categories** in (4): for example, we have not specified that the predicate is a verb, and the subject is a noun phrase. There is a cross-linguistic correspondence between syntactic categories, in that (say) most English nouns translate as nouns in French, and this is due to the semantic link that the names of physical objects are generally realised as nouns. There is a standard translation strategy of **transposition**² which involves change of syntactic category, but as a default (unless there is an indication to the contrary), categories are maintained in MT. It is a moot point whether syntactic categories are a help or a hindrance in abstract representations in MT: they can help to define and constrain these representations, but at the same time they will make transfer more complex if transposition is involved.

In (4) we have shown the **argument structure** of *mend*: it occurs with a subject and object. There is much linguistic research on argument structure and the kinds of abstract representation assumed here. We can say that many verbs require particular sets of arguments, e.g., a subject for *die*, and a subject, direct object and indirect object in the case of *give* (see (12) below). They also allow various extra modifiers, e.g., we could add *on Tuesday* to (3a).

The same analysis, (4), can be assigned to (5), the passive version of (3a).

(5) The sink was mended by a plumber.

The sink in (5) is the surface or grammatical subject, but would still be shown as the object in (4). “Subject” and “object” would now have to be interpreted in logical rather than surface terms, and one might want to add features “active” and “passive” to distinguish the two sentences, but the fact that (3a) and (5) are nearly synonymous would be captured. The words *was* and *by*, which are simply used to indicate a passive construction, are not shown at all in the abstract representation.

A representation such as (4) could be rendered more semantic (and hence more abstract) by replacing “subject” and “object” with terms like “agent” and “patient”, as is done in a number of MT systems. We could then go on to say that in (6), *John* is “experiencer”, rather than agent (see the beginning of Section 4).

(6) John liked the film.

Here, however, we prefer to leave (4) as it is, and to look at other aspects of these representations.

The sentences in (7) (among others) can equally be viewed as paraphrases of (3a).

- (7) a. It was a plumber who mended the sink.
b. What the plumber did was mend the sink.
c. It was the sink that the plumber mended.

The variations in such a paradigm can be abstracted away from and all such examples can be represented essentially as in (4), capturing the fact that all the examples deal with the same situation, of a plumber mending the sink. Of course, it has to be acknowledged that the examples are used in different contexts (see Section 6). Abstract representations, then, may involve **neutralization** of many surface differences. We could say that these are all different manifestations of the same **canonical form**. We can extend the paradigm by adding questions and negatives as in (8).

- (8) a. Did a plumber mend the sink?
b. A plumber did not mend the sink.

Features could be added to distinguish these from declaratives and positives, but it should be clear that they are realizations of the same argument structure. (Negative sentences raise extra translational problems, though; see Section 6.)

Consider also the examples in (9).

- (9) a. The plumber seems to have mended the sink.
b. The sink seems to have been mended by the plumber.
c. I believe the plumber to have mended the sink.
d. The plumber is believed to have mended the sink.

Each of these involves the same situation of sink-mending by a plumber, even though the words denoting these entities may be far apart in the sentence. However, the representation in (4) can usefully form part of the interface structure for the examples in (9) — embedded within a representation for the rest of the sentence, of course. Besides neutralization, then, abstract representations also involve the “undoing” of various kinds of syntactic processes, often described by linguists in terms of the metaphor of movement. For instance, in (9a) one might say that the phrase *the plumber* has been moved from the subordinate clause, where it logically belongs, to the main clause (cf. the paraphrase (10)).

- (10) It seems that the plumber has mended the sink.

These ideas — which are realized in a variety of ways in MT systems — are recognizably similar to the approach to human translation taken by Nida and Taber (1969), in which expressions are recast as “kernels”. These are strings of words,

rather than abstract structures, but essentially the same idea, of representing related expressions by a single form, is adopted. Nida and Taber (page 48) also ignore syntactic categories, as they would assign the same kernel to all the forms in (11). Certainly, interlingua-based systems will have to do something along these lines.

- (11) a. She sings beautifully.
- b. the beauty of her singing
- c. Her singing is beautiful.
- d. her beautiful singing

Such an approach has been criticized by translation theorists, but here we shall concentrate on its positive aspects. Its justification, for both human and machine translation, is as follows: “languages agree far more on the level of the kernels than on the level of the more elaborate structures [surface structures]” (Nida and Taber 1969: 39). That is to say, languages differ in terms of the variations from abstract forms that they allow. A simple illustration is that English allows two constructions with verbs of giving (12), while French allows just one (13).

- (12) a. Anne gave a book to Charles.
- b. Anne gave Charles a book.
- (13) a. *Anne a donné un livre à Charles.*
Anne has given a book to Charles
- b. **Anne a donné Charles un livre.*
Anne has given Charles a book

English *give* and its French equivalent *donner* require the same arguments at an abstract level, but their surface realizations differ. The same abstract structure is found in German, where the indirect object is in the dative case. Example (12) illustrates an **alternation** between two ways of realizing an abstract form, an alternation not found in French.

A more spectacular set of differences in terms of realizations of canonical forms can be found in a range of phenomena such as the English construction sometimes known as “Tough-Movement”, whereby the logical object of a subordinate clause is moved to be the surface subject of the main clause, as in (14).³

- (14) a. He is easy to convince.
- b. Linguistics is boring to study.

Example (14b), for instance, can be paraphrased as (15).

- (15) To study linguistics is boring.

German allows this construction, but it is far more restricted, and (16), a literal translation of (14b), is ungrammatical.

- (16) **Die Linguistik ist langweilig zu studieren.*
the linguistics is boring to study

Instead, a translation along the lines of (17) is needed.

- (17) *Es ist langweilig, die Linguistik zu studieren.*
it is boring the linguistics to study
'It is boring to study linguistics.'

Here the phrase *die Linguistik* 'linguistics' has been restored to its logical position as object of *studieren* 'to to study'.

The book by Hawkins is a good example of research in contrastive linguistics which has clear implications for translation. For instance, he develops the following generalization:

The set of German surface structures in which a phrase bears a surface grammatical relation to a predicate with which it has no logical relation is properly included in the corresponding English set.

This is taken from Hawkins (1986: 97) but has been reworded to make use of terms employed in the present chapter. Although Hawkins does not himself discuss translation, it is a short step to conclude that translating many English structures into German may be facilitated if a logical or canonical type of interface representation is built.

It should be noted that lexical disambiguation can also require restoration of logical relations. For instance, *adopt* has the two meanings 'take into one's family' and 'formally approve'. The ambiguity is sometimes maintained in translation (e.g. French *adopter*, German *adoptieren*), but not always (e.g. Danish *adoptere* vs. *vedtage*). The correct interpretation is usually dependent on the semantics of the grammatical object as in (18).

- (18) a. They adopted the proposal.
b. They adopted the child.

But now consider (19), where the logical object is far removed from the word *adopt*.

- (19) This proposal seems to have been adopted.

In (19), disambiguation of *adopt* requires reconstruction of the fact that *this proposal* is its logical object, thus permitting selection of the 'formal approval' sense. Mapping to more abstract forms, then, is helpful not just when the source surface structure cannot be employed in the target language, but also when lexical disambiguation is called for.

4. Translation divergences

It is often claimed that representations for MT should be more abstract than the ideas we have sketched in the previous section, which means that they should be more faithful to the *meaning* of sentences rather than to their syntactic *structure* (even an underlying structure). We have already touched on these ideas when we mentioned the omission of syntactic category information from abstract representations, and the use of labels like “agent” and “experiencer”. This latter may be useful in dealing with cases such as (20). The Italian verb *piacere* may be said to correspond to English *like*, but the subject and object are switched round. A representation along the lines of (21) would neutralize the differences.

- (20) *Roma mi piace.*
Rome me pleases
'I like Rome'
- (21) predicate: *like / piacere*
experiencer: *I / mi*
stimulus: *Rome / Roma*

The “experiencer” role here is linked to the logical subject in English, but to the logical object in Italian. It is sometimes claimed that cases where the experiencer is realized as object differ in that they contain an element of causation ('cause to enjoy') that is absent from the experiencer-as-subject examples. Be that as it may, it seems reasonable to take the *like–piacere* examples as translation equivalents.

In addition, it might be claimed that words should be broken down into more primitive semantic notions, e.g. *kill* might be decomposed into ‘cause to become dead’. However, this last idea is likely to complicate translation, rather than simplify it, as (say) *kill* could no longer be mapped straightforwardly to French *tuér*. Instead, *kill* would first have to be decomposed in analysis, and then the decomposition would have to be “reassembled” as *tuér* in synthesis.

We now concentrate on some more radical instances of source–target language differences, namely **translation divergences**:

A translation divergence is a difference in syntactic surface structure between sentences of the same meaning in different languages; the semantic content of the source language sentence is expressed by different morphological or syntactic forms in the target language. (Vandooren, 1993: 77).⁴

Examples are a commonplace in the MT literature; a standard example is the French sentence in (22a) and its English translation, (22b).

- (22) a. *Il traversa la rivière à la nage.*
he crossed the river at the swim
'He crossed the river swimming'
b. He swam across the river.

The kind of mapping to underlying form seen in the last section is not helpful here, as the verbs are not translation equivalents (*traverser* 'cross' vs. *swim*). In a transfer-based system, one could simply accept that this is a case where source and target language cannot be made to correspond, and where the burden of stating the structural and lexical discrepancies is to be borne by the transfer module.

However, interlingual — and many transfer — systems would not accept such a conclusion. The challenge then is to find some way of representing (22) so that the structures correspond more or less completely to each other, thus simplifying transfer. There is a sizeable linguistic literature on the way expressions of motion vary in their realization: Talmy (1985) classifies languages in terms of how they conflate the elements of a motion event into words. Romance languages, for instance, combine the idea of motion with the path followed into the verb; so *traversa* in (22a) shows that the path taken by the subject was across, and not (say) upstream. But other Indo-European languages combine motion with manner; *swim* conveys the manner of motion (swam as opposed, say, to jumped or drove). This is interesting, but does not in itself suggest what abstract representations should be like. Equally, translation-oriented discussions, such as Vinay and Darbelnet (1958: 105), are insightful, but do not really contribute to the problem we are currently addressing. Slobin (1996) shows that professional human translations from English to Spanish of sentences with motion expressions often involve omission of information or fairly drastic restructuring into two clauses (23)–(24).

- (23) a. He strolled across the room to the door.
b. *Se dirigió a la puerta.*
self he-went to the door
'He went to the door.'
- (24) a. Marta walked through the park and along the avenues.
b. *Marta cruzó el parque y paseó a lo largo de las avenidas.*
Marta crossed the park and promenaded at the length of the avenues.
'Martha crossed the park and walked along the avenues.'

But this kind of repackaging of information is generally not found in MT, which sticks more closely to source-language structure.

Dorr (1993) proposes an interlingual representation based on the ideas on

semantics of Jackendoff (1983). Example (22) would be analysed as in (25), here presented in an informal way.

- (25) event: *go*
 THING: *he*
 PATH: *to [across river]*
 MANNER: *swimmingly*

This resembles Talmy's idea of the elements of a motion event (see above), and is part of a larger semantic framework for the representation of spatial and non-spatial situations. But the idea that *swimmingly* could be a semantic primitive is very unattractive, and Dorr essentially uses Jackendoff's MANNER field as a catch-all — for example, the verb *read* involves a MANNER field with the value *readingly*, an analysis which is a hindrance, rather than a help, in translation.

We may conclude, then, that it is difficult to import linguistic ideas on abstract or semantic representations into MT wholesale. The linguistic proposals are not developed with translation in mind, and often have to be fine-tuned or extended in some way. The difficulty is in developing interlingual ideas which are not unconsciously oriented towards specific (types of) languages, but are still linguistically motivated, as opposed to *ad hoc* solutions to a particular translation problem. *Ad hoc* proposals cannot be extended to other, similar phenomena, and generally cannot cope with interacting problems in a single clause.

5. The translation of tense

In this section we present a case study in the treatment of an element of meaning, namely reference to time via the linguistic feature of tense. In (4) we represented tense by means of a feature "past" on the verb. But since, as we shall see, tenses frequently do not correspond cross-linguistically, we need to examine this part of language in more detail. A transfer system might be content with statements along the lines of "tense X is translated as tense Y in context Z, and otherwise as tense W". But interlingual systems, and many transfer systems, would need to propose some semantic representation of tense that could form part of abstract structures.⁵

Consider the German sentence in (26a) and the problem of how to translate it as (26c) as opposed to (26b). Example (26a) uses the German perfect tense, consisting of the verb *haben* 'have' plus past participle, but English needs the future perfect tense here (*shall have...*), as the present perfect (*have* on its own) is incompatible with reference to the future (as can be seen from (26b)). To build up a semantic representation for the temporal structure of (26a,c), we need to refer to

three distinct time points or periods. The first time is that at which the sentence is written or uttered, known conventionally as the “Speech Time” (abbreviated as S). The second time is that at or during which the event of settling everything takes place, the “Event Time” (E). The third time in (26) is *tomorrow* (the day after the day including S), to which the other times are related; it is the “Reference Time” (R).

- (26) a. *Bis morgen habe ich alles geregelt.*
by tomorrow have I everything settled
b. *By tomorrow I have settled everything.
c. By tomorrow I shall have settled everything.

The representation for (26a,c) involves specifying the temporal relations of both E and S *vis-à-vis* R (27).

- (27) a. E before R
b. R after S

Both S and E precede R: the speech time is before *tomorrow*, and so is the time of settling. S and E are not directly related, as (26a,c) do not specify whether the settling takes place before, at, or after the speech time: consider, for instance, that (26c) is true — though misleading — if the speaker has already settled everything at the time of speaking.

Now we must examine how the tense forms and tense representations can be mapped to and from each other. Synthesising the English forms should be relatively straightforward, as the form–meaning correlations in (28) hold, ordinarily.

- (28) a. E before R \leftrightarrow *have*
b. R after S \leftrightarrow *will*

Mapping (26a) to (27) is more problematic, though, as the German perfect tense has other uses (29).

- (29) a. *Ich habe ihn gestern gesehen.*
I have him yesterday seen
'I saw him yesterday.'
b. *Es hat geschneit. Wir können fahren.*
It has snowed we can leave
'It has been snowing. We can leave.'

So the German perfect tense corresponds not just to the English future perfect tense but also to the present perfect and simple past tenses. It is probably best to take the German form as ambiguous between two meanings (30)–(31).

- (30) a. E before R
- b. R not before S
- (31) a. E = R
- b. R before S

The structure in (30) covers the use in (26a) and (29b): it would be the equivalent of both the English future perfect tense (R after S) and the present perfect tense (R = S, meaning that R and S coincide). In contrast, (31) is for the simple past tense: the event is simultaneous with a reference time which precedes speech time.

The question now is how to choose the appropriate representation for (26a): how to discard (31) and choose a more precise form of (30). The key lies in the adverbial *bis morgen*. It seems reasonable to assume that *morgen* ‘tomorrow’ establishes an R that is later than S, and that this should be part of the information associated with the word in the lexicon — its lexical entry. The representation for (26a) is arrived at by combining the descriptions of the tense form and the adverbial and rejecting any contradictory analyses. So (31) is rejected because it specifies (31b) R before S, whereas *morgen* specifies R after S. (30) is compatible with the description of *morgen*, but since the latter is more precise that is the one selected. We then combine the R after S from *morgen* with E before R from *haben* to obtain (27).

We have in this section sketched how tense forms can be represented in an interlingual way and how the forms can be mapped to meanings. Note that this requires at least: (a) an adequate linguistic theory of tense, (b) adequate descriptions of the languages involved, and (c) ideas on how to use the information in a sentence to home in on the correct representation.

6. Discourse and function

Handbooks on translation⁶ emphasize that sentences cannot be dealt with in isolation, but must be seen as part of a text, and a well-formed text requires that the sentences forming it are related in an appropriate way. Moreover, texts exist within some non-linguistic context which includes the communicative goals of speakers and writers. There is no doubt that translations which fail to take textual structure into account can read, at best, very awkwardly.

For instance, the distinction between **given information** (assumed to be already known to the reader) and **new information** can be used to determine which constituent is to be the surface subject of a clause, with subjects generally representing given information. Or issues of contrast can determine what is to be **topicalized** or used as part of some “marked” structure. It should be clear, for instance, that the sentences in (7) above should be used in rather special circumstances, and will

generally not be alternatives to (3a).

But equally, the active–passive pair (3a) and (5) are not just used at random. The choice between them is driven by considerations of how they fit into their context. The passive is frequently seen as a means of topicalizing the logical object by making it the surface subject. And constructions which are formally similar may be functionally very different, so that it will not always be appropriate, from a discourse point of view, to translate a source-language passive as a target-language passive. It is often pointed out that English passives with no *by*-phrase may be most felicitously translated as actives with some kind of unspecified subject.⁷ As another kind of example, consider the variety of ways of translating (32) into some languages, depending on whether it is being implied that someone else stole the watch, or she stole another person's watch, or she stole something else belonging to John. The essential idea would be to map source-language **topic** into target-language topic, even if these were realized differently.⁸

- (32) She did not steal John's watch.

The problem is that concepts like “the function of the passive” or “the theme or topic of a sentence” are notoriously hard to define in a rigorous way. Here is one account of clause structure in functional terms:

The Theme is the element which serves as the point of departure of the message; it is that with which the clause is concerned. The remainder of the message, the part in which the Theme is developed, is called ... the Rheme. As a message structure, therefore, a clause consists of a Theme accompanied by a Rheme; and the structure is expressed by the order — whatever is chosen as the Theme is put first. (Halliday, 1985: 38)

So in (3a) *a plumber* would be **theme**, and *mended the sink* **rheme**. Representation (4) could be extended to indicate this (33).

- (33) predicate: *mend* (past)
subject: *plumber* (theme, singular, indefinite)
object: *sink* (singular, definite)

Although Halliday goes on to refine the notion of theme somewhat, it remains unclear how to determine where the theme ends, or just what is intended by the vague expression “the point of departure of the message”. Even more sophisticated accounts of theme, topic and focus remain very subjective.⁹ Fawcett (1997, Ch.8) expresses scepticism about the usefulness of such concepts to the human translator.

As the notions in question are difficult to define in such a way that human linguists can assign them accurately, they are even harder to implement computationally. Consequently, MT systems generally have little to say about text structure,

and are confined to the sentence as the largest translation unit. Existing work on discourse in MT is programmatic and restricted in scope.¹⁰ It is certainly possible to implement translation heuristics such as (34).

- (34) SL short passive \Rightarrow TL active with unspecified subject

This relies on bilingual contrastive knowledge but makes no direct reference to discourse notions such as topics.

We may conclude that this is an area where intuitive and somewhat subjective linguistic notions may be of some use to the human translator, but are as yet not in a position to benefit MT other than in a limited way.

It should be added, though, that discourse structure can also be relevant to identifying the antecedents of pronouns.¹¹ For instance, consider the sequence in (35), noting that for some languages it is necessary to identify what the word *it* in the second sentence refers to in order to translate it correctly.

- (35) The analysis shows that the effects of the acid can be mitigated. It will therefore be used in what follows.

Here the *it* in the second sentence is most likely to be referring to *the analysis*, rather than to *the acid*, and this can be captured by stating that normally (though not unexceptionally) a topic (however defined) is maintained across sentences. So the topic of the first sentence is most likely to be the topic of the second. Other more complex examples where discourse structure assists in determining pronominal antecedents can also be constructed, though again it is not hard to find counter-examples. But this is definitely an area where discourse considerations can contribute to an answer — one which will involve exploiting a large variety of information.

7. Conclusion

Fawcett warns against those who

...want to use linguistics as a recipe giving ready-made solutions to specific translation problems rather than as a resource for extrapolating problem-solving techniques from specific concrete problems. (Fawcett, 1997: 2)

This is in the context of human translation, but it applies to MT as well. In this chapter we have tried to show how linguistics can contribute to the development of MT systems, and indeed that it has to do so if significant progress in MT is to be made. Linguistics has not solved the problems of MT, but it can help the researcher to reach solutions, by offering a range of observations, techniques and theories that may be adopted and extended within the MT enterprise.

Further reading

A useful survey of the relevance of linguistics to *human* translation, covering far wider ground than is done here, can be found in Fawcett (1997). For a discussion of the relationship between morphology and syntax in MT, see Bennett (1993). Language typology is the subject of Comrie (1981) and Croft (1990). For a discussion of computational techniques of lemmatization see Ritchie et al. (1992). Alleganza et al. (1991: 26–37) discuss linguistic research on argument structure and abstract representation in an MT context. Levin (1993) provides a comprehensive study of grammatical alternations in English.

On the subject of translation divergences, in addition to Vandooren's paper, see Hutchins and Somers (1992: 103ff). More technical discussions can be found in Nagao (1987), Tsutsumi (1990) and Dorr (1993). See also Bennett (1994) and Hutchins and Somers (1992: 138). For assessment and criticism of Dorr's work in particular, see Arnold (1996).

On the topic of time and tense, see especially Comrie (1985). On discourse and pronominal reference, see Fox (1987) and Hirst (1981).

Notes

1. See Chapter 1 for an explanation of conventions used for linguistic examples in this book.
2. This term is from Vinay and Darbelnet (1958: 96).
3. This, and other similar contrasts between German and English, are described by Hawkins (1986).
4. “Une divergence de traduction est la différence de structure syntaxique de surface que présentent des phrases de sens équivalent dans des langues différentes; le contenu sémantique de la phrase en langue source [...] est exprimé par des formes morphologiques ou syntaxiques différentes en langue cible.” (My translation — PB)
5. There is an enormous literature on tense. In our discussion we have assumed the essentially standard notions presented, *inter alia*, in Comrie (1985). We also make use of the ideas on form–meaning mappings in Hornstein (1990), of the analysis for German proposed by Thieroff (1994), and of the discussion of MT treatment of tense in Alleganza et al. (1991: 37–68).
6. For example, Hatim and Mason (1990, 1997), Baker (1992), Hervey and Higgins (1992), Lonsdale (1996).
7. See Zhu (1996) on Chinese, for instance.
8. The example is from Bressan (1987).
9. For example, Lambrecht (1994).

10. Defrise (1994), Steiner (1994).
11. See Kehler (1993) among much other work on anaphora.

References

- Allegranza, Valerio, Paul Bennett, Jacques Durand, Frank van Eynde, Lee Humphreys, Paul Schmidt and Erich Steiner (1991) "Linguistics for Machine Translation: The Eurotra Linguistic Specifications", in Charles Copeland, Jacques Durand, Steven Krauwer and Bente Maegaard (eds), *The Eurotra Linguistic Specifications*, Luxembourg: Office for Official Publications of the European Communities, pages 15–123.
- Arnold, Douglas (1996) "Parameterizing Lexical Conceptual Structure for Interlingual Machine Translation", *Machine Translation* 11, 217–241.
- Baker, Mona (1992) *In Other Words: A Coursebook on Translation*. London: Routledge.
- Bennett, Paul (1993) "The Interaction of Syntax and Morphology in Machine Translation", in Frank Van Eynde (ed.) *Linguistic Issues in Machine Translation*, London: Pinter, pages 72–104.
- Bennett, Paul (1994) "The Translation Unit in Human and Machine", *Babel* 40, 12–20.
- Bressan, D. (1987) "Emphatic Devices in English–Italian" Translation, *Multilingua* 6, 69–75.
- Comrie, Bernard (1981) *Language Universals and Linguistic Typology: Syntax and Morphology*. Oxford: Basil Blackwell.
- Comrie, Bernard (1985) *Tense*. Cambridge: Cambridge University Press.
- Croft, William (1990) *Typology and Universals*. Cambridge: Cambridge University Press.
- Defrise, Christine (1994) "The Treatment of Discourse in Knowledge-Based Machine Translation", in Ramm (1994), pages 53–75.
- Dorr, Bonnie Jean (1993) *Machine Translation: A View from the Lexicon*. Cambridge, Massachusetts: The MIT Press.
- Fawcett, Peter (1997) *Translation and Language*. Manchester: St Jerome Publishing.
- Fox, Barabra A. (1987) *Discourse Structure and Anaphora*. Cambridge: Cambridge University Press.
- Halliday, Michael A. K. (1985) *An Introduction to Functional Grammar*. London: Edward Arnold.
- Hatim, Basil and Ian Mason (1990) *Discourse and the Translator*. London: Longman.
- Hatim, Basil and Ian Mason (1997) *The Translator as Communicator*. London: Longman.
- Hawkins, John A. (1986) *A Comparative Typology of English and German*. London: Croom Helm.
- Hervey, Sándor and Ian Higgins (1992) *Thinking Translation: a Course in Translation Method: French to English*. London: Routledge.
- Hirst, Graeme (1981) *Anaphora in Natural Language Understanding*. Berlin: Springer.
- Hornstein, Norbert (1990) *As Time Goes By: Tense and Universal Grammar*. Cambridge, Mass.: MIT Press.
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*.

- London: Academic Press.
- Jackendoff, Ray (1983) *Semantics and Cognition*. Cambridge, Mass.: MIT Press.
- James, Carl (1980) *Contrastive Analysis*. London: Longman.
- Kehler, Andrew (1993) "The Effect of Establishing Coherence in Ellipsis and Anaphora Resolution", in *31st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, Columbus, Ohio, pages 62–69.
- Lambrecht, Knud (1994) *Information Structure and Sentence Form: Topic, focus, and the Mental Representation of Discourse Referents*. Cambridge: Cambridge University Press.
- Levin, Beth (1993) *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago: The University of Chicago Press.
- Lonsdale, Alison Beeby (1996) *Teaching Translation from Spanish to English*. Ottawa: University of Ottawa Press.
- Nagao, Makoto (1987) "Role of Structural Transformation in a Machine Translation System", in Sergei Nirenburg (ed.) *Machine Translation: Theoretical and Methodological Issues*, Cambridge: Cambridge University Press, pages 262–277.
- Nida, Eugene S. and C. R. Taber (1969) *The Theory and Practice of Translation*. Leiden: E. J. Brill.
- Ramm, Wiebke (ed.) (1994) *Text and Context in Machine Translation: Aspects of Discourse Representation and Discourse Processing*. Luxembourg: Office for Official Publications of the European Communities.
- Ritchie, Graeme D., Graham J. Russell, Alan W. Black and Stephen G. Pulman (1992) *Computational Morphology: Practical Mechanisms for the English Lexicon*. Cambridge, Mass.: MIT Press.
- Slobin, Dan (1996) "Two Ways to Travel: Verbs of Motion in English and Spanish", in Masayoshi Shibatani and Sandra A. Thompson (eds), *Grammatical Constructions*, Oxford: Clarendon Press, pages 195–219.
- Steiner, Erich (1994) "A Fragment of a Multilingual Transfer Component and its Relation to Discourse Knowledge", in Ramm (1994), pages 77–115.
- Talmy, Leonard (1985) "Lexicalization Patterns: Semantic Structure in Lexical Forms", in Timothy Shopen (ed.) *Language Typology and Syntactic Description III: Grammatical Categories and the Lexicon*, Cambridge: Cambridge University Press, pages 57–149.
- Thieroff, R. (1994) "Perfect and Pluperfect in German", in Co Vet and Carl Veters (eds) *Tense and Aspect in Discourse*, Berlin: Mouton de Gruyter, pages 99–113.
- Tsutsumi, Taijiro (1990) "Wide-Range Restructuring of Intermediate Representations in Machine Translation", *Computational Linguistics* 16, 71–78.
- Vandooren, Françoise (1993) "Divergences de traduction et architectures de transfert", in Pierrette Bouillon and André Clas (eds) *La traductique: Études et recherches de traduction par ordinateur*, Montréal: Les Presses de l'Université de Montréal, pages 77–90.
- Vinay, Jean-Paul and Jean Darbelnet (1958) *Stylistique comparée du français et de l'anglais*. Montréal: Beauchemin; available in English as *Comparative Stylistics of French and English: A Methodology for Translation*, translated and edited by Juan C. Sager and M.-J. Hamel, Amsterdam (1995): John Benjamins.
- Zhu, C. (1996) "Syntactic Status of the Agent: Its Significance for Information Presentation in Translating the Passive between Chinese and English", *Multilingua* 15, 397–417.

CHAPTER 10

Commercial systems

The state of the art

John Hutchins

University of East Anglia, Norwich, England

1. Introduction

In a general overview of the availability and potential usefulness of commercial machine translation (MT) systems and translation tools, it is important to distinguish three basic types of translation demand: dissemination, assimilation, and interchange.

The first, and traditional one, is the demand for translations of a quality normally expected from human translators, i.e. translations of publishable quality — whether actually printed and sold, or whether distributed internally within a company or organisation. The use of MT for **dissemination** purposes has been satisfied, to some extent, by MT systems ever since they were first developed in the 1960s. However, MT systems produce output which must invariably be revised or post-edited by human translators if it is to reach the quality required. Sometimes such revision may be substantial, so that in effect the MT system is producing a draft translation. As an alternative, the input text may be regularised (or “controlled” in vocabulary and sentence structure — see Chapter 14) so that the MT system produces few errors which have to be corrected. Some MT systems have, however, been developed to deal with a very narrow range of text content and language style, and these may require little or no preparation or revision of texts (see Chapter 15).

In recent years, the use of MT systems for these purposes has been joined by developments in translation tools (e.g. terminology databases and translation memories — TMs), often integrated with authoring and publishing processes. These “translator’s workstations” (see Chapter 2) are more attractive to human translators. Whereas with MT systems they see themselves as subordinate to the machine, in so far as they edit, correct or re-translate the output from a computer,

with translator's workstations the translators are in control of computer-based facilities producing output which they can accept or reject as they wish.

The second basic demand is for translations at a somewhat lower level of quality (and particularly in style), which are intended for users who want to find out the essential content of a particular document or database resource — and generally, as quickly as possible. The use of MT for **assimilation** has been met as, in effect, a by-product of systems designed originally for the dissemination application, since some users found that they could extract what they needed to know (e.g. for screening and/or information gathering) from the unedited MT output. They would rather have some translation, however poor, than no translation at all. With the coming of cheaper PC-based systems on the market, this type of use (often known as "gisting") has undoubtedly grown substantially.

Related to this application is translation within multilingual systems of information retrieval, information extraction, database access, etc. Here MT systems operate as components of **information access** systems, i.e. translation software is integrated in other systems: (a) systems for the search and retrieval of full texts of documents from databases (generally electronic versions of journal articles in science, medicine and technology), or for the retrieval of bibliographic information; (b) systems for extracting information (e.g. product details) from texts, in particular from newspaper reports; (c) systems for summarising texts; and (d) systems for interrogating non-textual databases. As yet, however, there are few commercial systems available in this area.

The third type of demand is that for translation between participants in one-to-one communication (telephone or written correspondence). In this **interchange** use, the situation is changing quickly. The demand for translations of electronic texts on the Internet, such as e-mail and discussion groups, is developing rapidly. In this context, human translation is out of the question. The need is for immediate translation in order to convey the basic content of messages, however poor the input. MT systems are finding a natural role here, since they can operate virtually or in fact in real time and on-line and there is little objection to the inevitable poor quality.

Another context for MT in personal interchange is the focus of much research. This is the development of systems for **spoken language** translation, e.g. in telephone conversations and in business negotiations. The problems of integrating speech recognition and automatic translation are obviously formidable, but progress is nevertheless being made. In the future — still distant, perhaps — we may expect on-line MT systems for the translation of speech in highly restricted domains.

2. Types of systems

At the present time we may distinguish the following types of systems and their most appropriate areas of application:

- (a) mainframe, workstation and/or client-server systems on intranets of large organisations;
- (b) MT systems for professional translators;
- (c) translator's workstations for professional translators operating on company intranets or independently;
- (d) computerised translation tools: dictionaries, terminology management software, TM systems;
- (e) MT systems for occasional users and/or casual home use;
- (f) systems designed for Internet use and/or for translating web pages, either for company or individual use;
- (g) MT services on the Internet providing translations on demand.

Traditionally, MT systems have been divided according to architectures: direct translation, transfer-type, interlingua-based, statistics-based, example-based, etc., but these distinctions are largely irrelevant to and hidden from users, and they are ignored in this chapter. In general, however, it may be pointed out that commercial systems are based usually on well-tested approaches — for obvious reasons — and these tend to be based on the older traditional (linguistics rule-based) strategies developed from the 1960s to the late 1980s. More recent developments in MT research in the 1990s based on text corpora — the statistics-based and example-based approaches — have not yet had much impact on the commercial MT scene. Increasingly, there are however systems incorporating example-based methods, and of course the translator's workstations make considerable use of statistics-based facilities for the creation and utilization of TMs, i.e. bilingual corpora of previous translations and their originals.

As throughout the computing industry, there has been a *de facto* standardisation of hardware, operating systems and inter-compatibility. In particular, for the smaller systems, the standards are PC compatibles, Pentium CPUs, Microsoft *Windows 95, 98, ME, 2000, NT*, etc. A few are still available for Microsoft DOS systems, and some (although increasingly rarely) are designed for Macintosh equipment. As for Internet access, nearly all systems either include or run with *Netscape Navigator*, *Netscape Communicator*, or Microsoft *Internet Explorer*. MT products for Japanese, Chinese, and Korean generally require additional software (e.g. *Japanese Windows*, *Japanese Language Kit*), and occasionally run only on proprietary hardware.

The focus of this chapter will be the development and use of commercially available systems for dissemination, i.e. for aiding the production of “publishable” quality translations. Other applications will be treated more briefly. Changes in the MT market are very rapid: every year there are many new systems, many developments in old systems (new platforms, new languages, etc.), companies merge or cease trading, and many products become no longer available. Full details of systems available — including those mentioned here for illustrative purposes — may be found in the *Compendium of Translation Software* (see Further reading). This is a regularly updated listing of current commercial MT systems and computer-aided translation support tools (including translator’s workstations, terminology management systems, electronic dictionaries, localization support tools, etc.)

2.1 Mainframe, client-server and workstation systems

The oldest MT systems are those developed originally for mainframe computers, e.g. the *Systran*, *Logos* and *Fujitsu (Atlas)* systems. *Systran*, originally designed for translation only from Russian into English, is now available for a very large number of language pairs: English from and into most west European languages (French, German, Italian, Spanish, Portuguese), Japanese, Korean, etc. Likewise, *Logos*, originally marketed for German to English, was later available for other languages: English into French, German, Italian and Spanish, and German into French and Italian. The *Fujitsu Atlas* system, on the other hand, is still confined to translation between English and Japanese (in both directions).

Mainframe systems — much improved from their earlier 1960s and 1970s designs — are still available, primarily now it appears for large companies or organisations wanting to include an MT engine in already existing documentation systems, but for most purposes, large-scale systems take the form of workstations or client-server systems operating over company intranets. A popular choice for the workstation and/or server has been the Sun SPARCstation, and many of the older larger systems are still available for this platform. However, some Japanese computer companies chose to develop MT software for their own equipment, and some are still available commercially only on their proprietary platforms.

Needless to say, the prices of client-server systems make them affordable only for large companies or organisations with large translation services. From the mid 1990s onwards, most of these systems have begun to appear in cheaper versions for personal computers — although often with substantially smaller dictionary resources and without facilities for working in groups and networks.

The main customers and users of mainframe and client-server systems are the multinational companies exporting products and goods in the global market. The

need is primarily for translation of promotional and technical documentation. Technical documents are often required in very large volumes: a set of operational manuals for a single piece of equipment may amount to several thousands of pages. There can be frequent revisions with the appearance of new models. In addition, there must be consistency in translation: the same component must be referred to and translated the same way each time. This scale of technical translation is well beyond human capacity. Nevertheless, in order to be most cost-efficient, an MT system should be well integrated within the overall technical documentation processes of the company: from initial writing to final publishing and distribution. Translation systems are now being seamlessly integrated with other computer-based systems for the support of technical writers, not just assistance with terminology, but also on-line style manuals and grammar aids.

There are numerous examples of the successful and long-term use of MT systems by multinationals for technical documentation. One of the best known has been the application of the *Logos* system at the Lexi-Tech company in New Brunswick, Canada; initially for the translation into French of manuals for the maintenance of naval frigates, later as a service for many other large translation projects. *Systran* has had many large clients: Ford, General Motors, Aérospatiale, Berlitz, Xerox, etc. Users of *Logos* have included Ericsson, Osram, Océ Technologies, SAP and Corel. The *Metal* German–English system (no longer on the market) has been successfully used at a number of European companies: Boehringer Ingelheim, SAP, Philips, and the Union Bank of Switzerland.

A pre-requisite for successful MT installation in large companies is that the user expects a large volume of translation within a definable domain (subjects, products, etc.), and that the user has available (or has the resources required to acquire or to create) a terminological database for the particular application. The creation of terminology databases and the maintenance of large dictionaries demands considerable initial and continuing expenditure, which can usually be justified only if translation production is on a large scale. In fact, it is always desirable for company documentation to be consistent in the use of terminology. In addition, many companies insist upon their own use of terms, and will not accept the usage of others. To maintain such consistency is almost impossible outside an automated system. However, it does mean that before an MT system can be installed, the user must have already available a well-founded terminological database, with authorised translation equivalents in the languages involved, or — at least — must make a commitment to develop the required term bank.

Most large-scale MT systems have to be customised, to a greater or lesser extent, for the kind of language found in the types of documents produced in a specific company. This customisation may embrace the addition of specific grammatical rules to deal with frequent sentence and clause constructions, as well as the

inclusion of specific rules for dealing with lexical items, and not just those terms unique to the company. A further step is the implementation of a company-specific controlled language, not just for standardisation but for reducing well-known problems of MT such as lexical and structural ambiguities in source texts. The amount of work involved in such customisation and in the pre-editing control of input may not be justifiable unless output is in a number of different languages.

Large savings are reported by many companies that have installed MT systems: in some cases there have been reductions in the costs of producing finished translations of up to 40% or 50%, and nearly all companies report much faster throughputs. However, it must be stressed that it is only the larger organisations dealing with 100,000 pages a year or more that can expect such dramatic savings. Smaller companies and translation services may gain only in terms of speed of production and may experience few cost savings. The situation is, however, changing rapidly, and cheaper more powerful MT systems, combined with cheaper and more powerful publishing and authoring systems, will probably bring comparable savings to a wider range of companies and services.

Multinational companies at many locations in different countries are often linked by internal networks (intranets). In this environment, translation jobs can be passed easily in electronic form from one office or branch of the organisation to another. Indeed, a document may be authored in one location, sent for translation at another, and printed and distributed at a third. There are a number of client-server systems on the market, e.g. *Atlas* (from Fujitsu), *Systran Enterprise*, *Enterprise Translation Server* (SDL), *TranSphere* (AppTek), and *TransSmart* (Kielikone) and there are also companies that develop client-server software for specific customers, nearly always large government organisations or multinational corporations. The oldest is Smart Communications Inc. of New York, which has built systems for Ford, Citicorp, Canadian Department of Employment and Immigration, etc. Other companies include ESTeam Inc. of Greece, and LANT n.v. of Belgium

2.2 Translator's workstations

In the 1990s, the options for large-scale computer-based translation production broadened with the appearance on the market of translator's workstations (see Chapter 3). These combine multilingual word processing, means of receiving and sending electronic documents, facilities for document scanning by OCR (optical character recognition), terminology management software, facilities for concordancing, and in particular TMs. The latter facility enables translators to store original texts and their translated versions side by side, so that corresponding sentences of the source and target are aligned. The translator can thus search for a phrase or even full sentence in one language in the TM and have displayed corre-

sponding phrases in the other language. These may be either exact matches or approximations ranked according to closeness (see Chapter 4).

It is often the case in large companies that technical documents, manuals, and so on undergo numerous revisions. Large parts may remain unchanged from one version to the next. With the TM, the translator can locate and re-use already translated sections. Even if there is not an exact match, the versions displayed may be usable with minor changes. Translator's workstations also give access to terminology databases, in particular to company-specific terminology, for words or phrases not found in the TM. In addition, many translator's workstations are now offering full automatic translations using commercial MT systems. The translator can choose to use them either for the whole text or for selected sentences, and can accept or reject the results as appropriate.

The translator's workstation has revolutionised the use of computers by translators. Translators have now a tool where they are in full control. They can use any of the facilities, or none of them, as they choose. As always, the value of each resource depends on the quality of the data. As in MT systems, the dictionaries and terminology databases demand effort, time and staff resources. TMs rely on the availability of suitable large corpora of authoritative translations — there is no point in using translations which are unacceptable (for whatever reason) to the company or the client.

2.3 Localisation support tools

One of the fastest growing areas for the use of computers in translation is software localisation (see Chapter 5). Here the demand is for producing documentation in many languages to be available at the time of the launch of new software. Translation has to be done quickly, but there is much repetition of information from one version to another. MT and, more recently, TMs in translator's workstations are the obvious solution. Among the first in this field was the large software company SAP AG in Germany, using older MT systems, *Metal* and *Logos*. Most localisation, however, is based on the TM and workstation approach — mainly *Transit*, *Déjà Vu*, and the *Trados Workbench*.

Localisation companies have been at the forefront of efforts in Europe to define standardised lexical resource and text-handling formats, and to develop common network infrastructures. The need for a general translation and management support environment for a wide variety of TM, MT and other productivity tools is seen as fundamental, and a number of companies are producing “localisation support tools”, for managing and routing localisation among translators, software engineers, project managers, for efficient use of different tools during overall processes, for automated updating, unified file tracking, etc.

The translation and management requirements of software localisation have been sufficiently distinct for the creation of a dedicated organisation (Localisation Industry Standards Association, LISA), which holds regular seminars and conferences throughout the world.

2.4 Systems for independent professional translators

For the independent translator, the translator's workstation may be no more affordable than the larger MT systems. Professional translators not employed by large organisations have currently two options: (a) relatively powerful systems capable of running on widely available computer equipment, e.g. *Windows*-based PC systems, and (b) translation support tools such as terminology management systems and TM programs.

Most vendors of client-server systems also have systems on the market designed primarily for the demands of the professional translator user, i.e. systems that have facilities for post-editing and publishing, and that can be used with terminology databases and sophisticated word-processing facilities. In origin, these systems are either downsized versions of mainframe (or client-server) systems or enhanced versions of cheaper PC systems. In the case of the former, often the same range of languages is covered as for the larger intranet versions, e.g. *Systran Professional*, and the two systems from the Pan American Health Organization (*Spanam* Spanish–English, and *Engspan* English–Spanish). What these systems lack in comparison with the intranet client-server systems are generally the wide range of document formatting and conversion facilities and sometimes the complete range of text-processing compatibility. However, even this situation is changing as standalone computers become more powerful, and as users' demands become clearer, so that increasingly these "professional" systems for the independent translator are acquiring the range of facilities found previously only in the largest mainframe and client-server systems.

2.5 Translation support tools

Just as large companies may well prefer translator's workstations to fully-fledged MT systems, the individual professional translator may not want to purchase an MT system that may cover only some of the languages required. Since the mid 1980s there has been a wide range of translation aids, some designed originally for workstations in larger organisations, intended primarily for individual translators for use on PC-type equipment.

Electronic dictionaries (usually in CD-ROM form) are available from nearly all

dictionary publishers, and from many companies supplying computer software. There are also many dictionaries accessible on the Internet.

Terminology management software provides facilities for professional translators to create, update and revise their own lexical resources, whatever the languages concerned (see Chapter 4). Typical facilities include means for downloading from on-line or other electronic databases. Software for TMs in individual packages (as opposed to components of translator's workstations) is being marketed by a number of vendors. These programs allow individual professional translators to build their own stores of searchable and aligned bilingual databases of original texts and their translations. Most can cope with texts in any language written in Roman characters, and some with non-Roman scripts.

2.6 Systems for non-professional (home) users

The basic need of the non-professional user of translation software is primarily as a means of access to foreign-language texts, to find out what a text in an unfamiliar or unknown language is about. What matters is the message. It is usually not essential to have a "perfect" translation. Any of the systems already mentioned can serve this need; indeed in earlier years one of the main uses of mainframe MT systems was the provision of rough translations, i.e. the unedited crude output, for the purposes of intelligence analysis or for scientific and technological reviews. At the European Commission, one of the principal uses of the *Systran* system is still the production of crude (sometimes lightly edited) translations for rapid surveys of documentation.

Software for personal computers began to appear in the early 1980s in systems from ALPS and Weidner. Their output was at a level of quality suitable only for information assimilation use, but they were too expensive for the casual home user. In fact they were bought mainly by professional translators, who found them frustratingly unsuited to their needs. This experience may have convinced professional translators that PC translation software would always be useless for their purposes, but the more recent "professional" systems described above are changing this perception.

It was not until PC equipment and software were much reduced in price during the early 1990s that this large potential "non-professional" market was opened up. Earliest in the field were the Japanese computer companies, selling systems, usually for English–Japanese and vice-versa, and designed to run on their own microcomputers. In the United States the earliest vendors were Linguistic Products with its series of *PC-Translator* systems, and Globalink, with its well-known *Power Translator*. They have been succeeded by numerous other vendors, many surviving only a

few years in this very competitive market. Many of the producers of client-server systems have sold versions of their systems for the home or non-professional market, but not always with the same large range of language pairs.

Finally, it may be noted that there is a proliferation of particularly inexpensive products, marketed as “translation systems” but which in fact are little more than electronic dictionaries. They sell presumably because of the widespread belief among those unfamiliar with translation that all that is needed in order to translate something is a bilingual dictionary.

Sales of PC translation software showed a dramatic rise during the 1990s. There are now estimated to be some 1,000 different MT packages on sale (when each language pair is counted separately.) For example, in Japan one system (*Korya Eiwa*, for English–Japanese translation) was said to have sold over 100,000 copies in its first year on the market. A recent development for many home-use systems has been the addition of facilities for voice input and voice output — this is not, of course, true translation of spoken language (conversation, etc.) but speech-to-text conversion, text-to-text translation, and text-to-speech synthesis.

Though it is difficult to establish how much of the translation software sold in large numbers is used regularly after initial purchase (some cynics claim that only a very small proportion is tried out more than once), there is no doubting the growing demand for “occasional” translation, i.e. by people from all backgrounds wanting gists of foreign text in their own language, or wanting to communicate in writing with others in other languages, however poor the quality. It is this latent market for low-quality translation, untapped until very recently, which is now being exploited. As a consequence, many products have to be treated with caution — in fact, they may not even meet minimal standards for crude “information only” translation.

2.7 MT for the Internet

The largest area of growth for translation demand is now undoubtedly based on use of the Internet. This is the need of the occasional user for software to translate web pages, e-mail and other Internet resources and texts, either off-line or on-line, and the availability of on-demand Internet-based translation services for companies.

There has been a rapid increase in MT software products designed specifically for online translation of web pages. Japanese companies such as Fujitsu, Toshiba, Hitachi and NEC have led the way, primarily with systems for translating from English into Japanese. They were followed quickly elsewhere, and nowadays, nearly all systems for home users incorporate web-page translation as standard features.

Equally significant has been the use of MT for e-mail and for “chat rooms”, many of the online systems mentioned having facilities for this application. In

addition, most home-use software is designed for this use, and some are specifically for e-mail and/or specifically for chat.

As well as these online systems there are now many Internet services offering translation facilities, many of them free. One of the earliest and probably still best known example is the AltaVista translation service *babelfish* (see Chapter 12). There have been many followers, although some offer not full translations but little more than on-line bilingual or multilingual dictionaries. The latter are undoubtedly serving a real need; even the use of the AltaVista service is apparently mainly for translating individual words or short phrases. When translations from on-line services are of complete sentences, the output is often poor. None of the systems has been designed specifically for translating the kind of colloquial, jargon-filled, and often “ungrammatical” language found in e-mail and on-line discussion forums.

At the same time however, there are now many network-based translation services for on-demand professional-level translation, generally with human revision as an option. In some cases these are client-server arrangements for regular users; in other cases, the service is provided on a trial basis, enabling companies to discover whether MT is worthwhile for their particular circumstances and in what form. In most cases, clients have the option of receiving unedited translations or versions post-edited by the suppliers’ own professional translators.

In the future, we may expect many more online translation services. There will be both a wider range of languages and a wider variety of charging methods. We may also expect to see services designed for particular domains and subject areas, since systems restricted to specific subjects have typically produced better quality output than general-purpose systems. Users of online translation systems (whether charged or free) will expect continued improvements, and this will be more likely with specialised services than with non-specialised ones.

2.8 MT for information access

The growing use of the Internet is highlighting the need for systems that combine translation with other language-oriented facilities, in particular database searching, information retrieval and summarisation. As yet, however, there are few such systems available commercially. Most of the web-page translators could be used for this purpose, although few enable search terms to be formulated and translated before searching the World Wide Web.

It is to be expected that in future this will be one of the main growth areas. Several research projects supported by the European Union combine MT with programs for information access, information extraction, and summarisation. There is equally intensive attention to this area in North America, in Japan, Korea,

China and other Asian countries. Many companies are directing their efforts to the development of products for the information marketplace.

3. Language coverage

From the very beginning of the commercialisation of MT systems, the major European languages have been well covered. Translation from English into French, German, Italian, Spanish, and Portuguese, and from these languages into English, is available from all the main vendors, and in most cases with versions for large organizations, as client-server systems, for professional translators, for home users, and for web-page and e-mail translation. In some cases, products are dedicated to particular pairs, e.g. German–English and Spanish–English.

Systems for other European languages are less common. The Scandinavian languages are relatively poorly covered, and although Russian was the main focus of the earliest MT research, there are now fewer products for this language (and for other Slavic languages) than for western European languages. Other languages of Europe have so far been neglected by the main vendors; there are no “professional” quality translation systems for Greek, Hungarian, Rumanian, Serbian, Catalan, or any of the Celtic languages.

In the 1980s, nearly all Japanese computer companies began marketing MT systems, predominantly between English and Japanese. In recent years, many more systems have appeared, a large number specifically for Internet/web use, which are obviously meeting a great demand in Japan. The older mainframe or workstation systems are now marketed also in *Windows* versions for either English to Japanese and/or Japanese to English, and almost every month there appears a new inexpensive system for translation between these two languages. But Japanese–English products come not only from Japanese companies; there is competition also from companies of US origin.

In contrast to Japanese, there are still few Chinese–English systems of reasonable quality, and most systems are intended for primarily non-professional use (interactive composition). The situation is slightly better for Korean–English, with some good-quality enterprise systems, although there are also many low-quality systems.

Other languages are even more poorly served. There have been surprisingly few systems for Arabic, despite the obvious potential market, and only one Hebrew–English system is marketed at present. While there have been systems for some African languages (by EPI-USE Systems (Pty) Ltd., South Africa), there are many languages still not covered by commercial systems, e.g. Indonesian, Malay, Viet-

namese, Thai, and languages of the Indian sub-continent (not even Hindi, Urdu and Bengali).

In principle, most translator's workstations are designed for use with a wide range of languages; they do not need programs for linguistic analysis and synthesis, only for dealing with strings of characters and words. However, the need for greater sophistication in the alignment programs of TM systems makes them less suitable for some languages than others, particularly non-European languages. However, although designed initially and primarily for languages using Roman alphabets, workstations are increasingly available in versions suitable for use with languages such as Arabic, Chinese and Japanese.

4. Conclusion

After many years of development, commercial MT systems are now capable of serving well the demands of multilingual companies and professional translators seeking cost-effective production of good-quality translation for dissemination purposes. This is particularly the case for translation between the major languages of the global marketplace. There remain many gaps for "minor" languages, including those of eastern Europe, Africa, and India.

Systems for assimilation purposes (for the less-demanding "occasional" user) are also widely available, with good language coverage on the whole. However, these systems often give poor-quality output, even for well-written source texts, let alone the low-level writing on e-mail and other Internet applications. There is clear need for improved quality in this area of commercial software, and even more for some consumer guidance in order that potential purchasers are not misled by exaggerated claims.

Further reading

The *Compendium of Translation Software* is available online at the website of the European Association for Machine Translation (www.eamt.org). Older editions of the *Compendium* — for tracking changes and developments — can be seen on the current author's website (<http://ourworld.compuserve.com/homepages/WJHutchins/compendium.htm>). For earlier surveys of MT systems see also Hutchins (1996, 1999, 2002).

References

The following articles are all available on the author's website:

<http://ourworld.compuserve.com/homepages/WJHutchins/>.

Hutchins, J. 1996. "Computer-based Translation Systems and Tools", *ELRA Newsletter* 1(4), December 1996, 6–9.

Hutchins, J. 1999. "The Development and Use of Machine Translation Systems and Computer-Based Translation Tools", in *International Conference on Machine Translation & Computer Language Information Processing*, Beijing, China, pages 1–16.

Hutchins, J. 2002. "The State of Machine Translation in Europe and Future Prospects", available on: <http://www.hltcentral.org>.

CHAPTER 11

Inside commercial machine translation

Scott Bennett and Laurie Gerber*

Denville, NJ / Language Technology Broker, San Diego, CA

1. Introduction

If you had to build an automatic translation system, assuming you knew how to do some computer programming, where would you start? How would you go about capturing and codifying all of the many levels of your knowledge about the grammars of the individual source and target languages you work with? How would you represent the many rules you use for mapping words, phrases and various grammatical constructions between source and target languages? How would you encode all of your knowledge about the way that different classes of words behave? Or about the idiosyncrasies of individual words? How would you keep all of that information organized? You make use of this information, and much more, every day as a translator.

Add to the challenges above the problem of making world knowledge — the part of communication that is not encoded in the message — available to computers, and you have the Gordian knot that linguists and computer scientists in both the commercial world and academia have been poking at for the last 50 years. These formidable programming and knowledge-engineering tasks continue to be at the heart of MT development efforts. They are the “hard” problems that we can only chip away at. However, commercial MT is not only a theoretical or engineering problem to be solved. The translation engine represents the hard problem for developers, but in order to make a system useful, other components are needed, for example user interfaces and tools to accompany, and facilitate work with, the MT system. In this chapter, we will explore a number of issues related to MT system development around two phases of the life of an MT system. First, how is it created? And second, what is ongoing relationship between developers and users?

Note that throughout this chapter we assume that the application of MT is to written text. This reflects two biases:

- (a) Historically, MT has been most often, and most successfully, applied to relatively formal or technical/scientific writing.
- (b) It is assumed that the readership of this book is interested primarily in the translation of written text.

2. Birth of an MT system

There are a variety of theoretical approaches to each of the steps in the translation process: the analysis of grammar, the encoding of lexicons, and resolution of ambiguity. No two MT systems employ exactly the same approach. However all of them have at least one thing in common: the need to maintain a coherent store of information about language, and rules for its analysis, translation and generation. Any established MT company must have a method for predictable, repeatable development of new language pairs. This does not imply that methods and technologies used by MT developers do not evolve, but that a production-quality MT system cannot be developed by an *ad hoc*, design-as-you-go process. In addition, the methodology must be able to scale up to manage the complexity and interactions of information about the whole language, not just tidy representative examples. The methodology by which an MT system handles these areas constitutes its distinguishing proprietary features, and forms the basis of the toolkit used by its development team.

Methods for building MT systems may be classified by their position on a continuum between two extremes: (a) Manually created systems where the lexicon, grammar and translation rules are written by linguists. We will call these “rule-based” systems. (b) Systems where patterns are learned automatically by the computer from texts. We will call these “data-driven” systems.

Virtually all commercial MT systems available at this writing (2002) are located at or very near the rule-based extreme. The description of rule-based MT system development in the next section roughly characterizes any of the commercial MT systems seen so far. However, data-driven systems have now been under development in research labs for over ten years, and some should emerge commercially by 2003 (cf. Knight, 1997). Among these emerging systems will be some that learn wholly automatically, and many more that are hybrids, adopting some mixture of manual development with automatic learning. The categories will continue to blur as traditionally rule-based MT developers also incorporate data-driven methods.

Regardless of the approach, it needs to be able to account for the orthography and grammar of any language it will be applied to. Sometimes limitations in the number and type of language pairs available from a developer reflect limitations in the generality or extensibility of their approach.

2.1 What's the big deal with developing new language pairs?

2.1.1 Rule-based MT development

Rule-based MT developers have internally defined proprietary grammars, and **symbolic representations**. The grammar allows linguists to catalog the types of linguistic phenomena that the system needs to use. When planning an MT system for a new language pair, the job of linguists and engineers is to identify appropriate mappings and parsing techniques between the set of phenomena realized in the new source and/or target languages and the system's grammar. The symbolic representation is the data structure in the computer that holds all of the grammatical information about a unit of text, and allows the parser to add incrementally new information as it is discovered, and query the information already stored. The unit of translation is usually a sentence.

For example, Table 1 is a highly abstract “data structure” showing how information might be added incrementally during the analysis process, the first phase of the translation process in a system that does multi-level grammatical analysis. The sentence being analysed is (1).

- (1) Click the start button with the mouse.

2.1.2 Information resources

One of the first challenges encountered when developing a rule-based MT system is where to find the resources — grammatical information about the languages involved, example texts for translation, lists of words and terms, and reliable translation equivalents for words and phrases. General-purpose systems, such as *Logos* and *Systran*, may be used on any type of text from any domain. This means that these systems must come equipped with a large general vocabulary, and that development work for production use must be grounded in observation and testing of

Table 1. Abstract data structure for sentence (1)

Word	Phase 1 information stored	Phase 2 more information	Phase 3 more information
Click	functions as verb	verb is imperative	verb has a direct object
the	functions as determiner	modifies <i>button</i>	
start	functions as noun	modifies <i>button</i>	
button	functions as noun	direct object of <i>click</i>	means ‘mechanical/icon’ (not ‘clothing fastener’)
with	instrumental preposition	adjunct of <i>click</i>	
the	functions as determiner	modifies <i>mouse</i>	
mouse	functions as noun	object of preposition <i>with</i>	means ‘computer device’ (not ‘rodent’)

extensive real-world text. Linguists typically work with source-language data, building dictionaries and writing grammar rules for the patterns they observe. The linguist's goal is to derive general rules from specific instances observed. The more textual data they analyse the better, though there are limits to the amount of data that linguists can effectively work with.

One might expect that grammar books or language-teaching texts would be the best resources for rules about language analysis. But it seems that when looking through real, *naturally occurring* text, there is no such thing as a “textbook example”, so grammar information for parser development comes from a variety of sources, ideally from large quantities of real text that is representative of the type of text the system will ultimately be used to translate. Published grammar reference books *may be* of some help, but their usefulness is hampered by two problems:

- Oversimplification: Grammar books for language learners tend to include a limited subset of grammatical constructions and features that can be easily generalized. In addition, they tend to focus primarily on spoken language phenomena, ignoring the grammar of written language.
- Overspecification: Grammar books written by linguists to catalog all features of a language tend to be so inclusive as to hide the prominent, common phenomena in a forest of detail and exception. These also tend to focus on spoken, rather than written language.

Thus, the development process must begin with examination of text samples and possibly construction of a basic text corpus that is representative of the type of text for which the system will most likely be used. The development team must establish conventions for representing the new language in the symbolic representation used, as described above (Yang and Gerber, 1996). For the developer who has a well-established symbolic representation into which linguistic and lexical data can be encoded, development is just a matter of time and effort.

Development begins by putting together a “shell” or framework of the system. This may initially be a series of empty program modules. A work plan is developed by breaking down and prioritizing the individual tasks within the parser, transfer and generation modules. At the same time, lexicographers begin building dictionaries, typically from frequency lists derived from the example texts. It is also sometimes practical to work from bilingual dictionaries, or (more innovatively) to try extracting translation equivalents from existing human-translated texts.

2.1.3 *Development of data-driven MT systems*

Data-driven approaches use machine-learning algorithms to capture automatically translation patterns. Systems called “example-based” or “statistical” fall into this category. Some data-driven MT systems use machine learning to emulate each step

of the process described above, including analysis, transfer, and synthesis. At the other extreme are purely statistical systems that eschew linguistic analysis *per se*. Such systems rely on algorithms that can learn correspondences between words and phrases in existing translations without worrying about grammar. For example, such a system will observe, in existing French–English translated data, that most of the time when the word *maison* appears in French, the word *house* appears in English. The system records this correspondence together with the probability that it occurs. Such systems can also learn the contextual conditions under which alternate translations should be used.

The strengths of data-driven systems are that they overcome human bias in making observations about how language will be used. The training/learning process is largely, if not wholly, automated and can save much of the time and effort of development and customization. Customization, discussed in Section 4.2 below, is very important to the success of many MT deployments. The weakness of data-driven approaches is that they require significant amounts of data to learn to translate general text. 1 million bilingual sentence pairs has been suggested as a good size for a training set for general purpose MT. Statistical approaches also tend to be very computer-resource intensive, requiring powerful processors and plenty of memory to translate in real time, as Figure 1 illustrates.

The “translation rules” learned by statistical systems consist of “parameters”, cross-lingual correspondences between words or phrases, accompanied by the probability that the word or phrase in the source language will be rendered as the word or phrase in the target language. In order to build such a system, sentence-to-sentence correspondences must be established, and words separated from punctuation, or “tokenized”. It is this aligned, tokenized “parallel corpus” that a statistical system learns from.

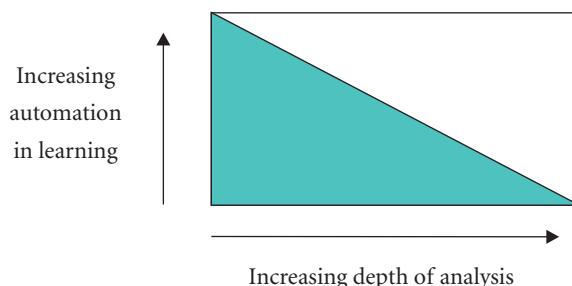


Figure 1. Typically, the greater the degree of automation in system development (learning of analysis and translation rules), the shallower the analysis the system performs. In the extreme case, learning is fully automated, and the system uses no conventional grammar or lexicon.

In building rule-based systems, we noted that appropriate dictionary coverage is crucial to translation quality. In building data-driven systems, it is important that the training material be representative of the text to be translated so that the learned parameters, which are analogous to the dictionary, contain the necessary terms.

2.1.4 How are languages targeted for development?

Development of translation systems so far has been an extensive, multi-year undertaking. Typically, each language pair is the work of several people requiring at least 1½ to 3 years for a commercial-quality release of a rule-based system. Automatically learned systems can be developed more quickly where training material is available, and we should see some of these emerging in the next few years. In addition to the initial cost of creating the system, maintenance and support are necessary for as long as the system is distributed to users. The high costs of development and maintenance are important factors limiting the language pairs considered for development. Because of the high initial investment required and long time to market, there is considerable commercial risk inherent in the development of translation systems for new languages. The tendency of MT developers in the USA to focus on only a few of the thousands of languages in the world reflects the political and economic interests of the user-base for MT. This small group of languages, typically Arabic, Chinese, French, German, Italian, Japanese, Portuguese, Russian and Spanish, represent the speaker communities with whom the USA has had the most trade, economic competition, and conflict. For example, economic growth, as well as conflict, in Korea and conflict in the former Yugoslavia have recently motivated new language-pair development. Ongoing economic and political relationships, for better or worse, guarantee an ongoing need for translation.

2.2 When is it ready? ...Ready for what?

Presumably, all MT developers have, as do Logos and Systran, internally determined performance thresholds for product release. Preparation for release includes an objective evaluation of system output: either on a targeted task (if the system is developed for a particular domain or text type), or on a balanced corpus representing various text types (if the MT system is intended to be a “general purpose” system).

The type of use a system is targeted for includes whether it will be primarily applied to

- assimilation or gisting tasks (information gathering, and browsing, where speed, and broad lexical coverage are more important than quality),
- dissemination tasks (translation for publication, where quality is most impor-

- tant, but the user has authoring control and may employ controlled language or at least work with a limited vocabulary and text type), or
- communication tasks (real-time e-mail translation, for example, where speed and accuracy are both important, as is the ability to handle informal language, but where extensive technical terms are unlikely to appear).

Note that these three tasks make very different demands on the grammar and lexicon of an MT system. Although they may appear to represent ascending stages of development, this is not the case. Developers may specialize in one approach, or may target different approaches for different language pairs. In the USA it is common for the language pairs *into* English to be primarily used for assimilation (commercial, academic, and governmental research and intelligence gathering), while the language pairs *out of* English are primarily used for dissemination (often translation of product literature or localization of software to speed penetration of overseas markets with exports.) All of these factors about the intended use of the system must be taken into account when assembling an evaluation test set of data for performance testing.

3. Performance

The work of developing an MT system is never finished. In fact, no MT system has achieved the goal that has been the holy grail of the MT world: FAHQT (Fully Automatic High Quality Translation), where “high quality” implies something approaching that of a good human translator for unrestricted input text. Methodological and technical advances have nudged up the quality ceiling for MT each year, but no one considers MT a solved problem. The highest performance achieved by MT systems is still in “sublanguage” applications where texts with relatively predictable style and vocabulary are to be translated, and the system can be trained to translate just that type of text (see Chapter 15). Although there are many useful applications for sublanguage translation, it is not representative of the larger body of naturally occurring language.

Some of the reasons for poor performance are described elsewhere in this book: Translation is a complex task even for human beings; it is much more difficult for computers. When researchers cite reasons why MT is a hard problem, they give examples of ambiguity and world knowledge that are hard for even human readers to understand in some cases. However, given that relatively consistent technical text has been the bread-and-butter application of MT, many of the phenomena that are stumbling blocks to high-quality output in practice are much more mundane than the theoretical imponderables cited by academics, such as the sentences in (2).

- (2) a. The box is in the pen.
b. Time flies like an arrow.

Common-sense knowledge is required to understand that *pen* cannot have the sense of ‘writing instrument’ in (2a). In (2b) there is surprising potential for misinterpretation by the naïve computer. For example, *time flies* could be a kind of fly that just happens to like arrows.... Often the mundane but crucial problems come down to lexical coverage of multi-word terms, prepositional-phrase attachment and preposition translation. Examples are (3a), where the words must be translated as a unit, not just the sum of the parts, and (3b) which can mean ‘wheel with six bolts’ or ‘attach ... with six bolts’.

- (3) a. butterfly nut
b. Attach the wheel with six bolts.

In the next section, we address some of the practical strategies that developers and users employ to improve the quality of MT output.

3.1 Long-term development strategy

3.1.1 *Text type and domain*

MT systems codify information about language in a static form. The rules work, and the lexical entries are valid for text that the system was developed for, and tested on, at a particular point in time. Some of this information is in fact reliably static. The basic grammar of a language evolves only very slowly. There are established vocabulary and writing styles for certain text types, for example patents, business letters, and newspaper articles. These also become conventionalized, and while not static, a style contains features that an MT system can be usefully trained for. We might think of domain and text type as forming two sets of parameters, where each possible pairing of parameters represents the characteristic vocabulary and grammatical style for a particular register within a particular domain. For example, a company that deals in computer software may have a need to translate correspondence, patents, technical manuals, help files, and marketing materials all incorporating computer nomenclature. A company that sells chemicals will have a similar range of text types all incorporating chemical nomenclature. Most MT systems have at least some of these parameters built in so that the user can select them when running a translation to improve output.

3.1.2 *Lexical change and innovation*

The lexicon of any language is in a constant state of flux. Words suddenly become popular or gradually fall out of use. Words evolve, taking on new grammatical

roles. Mercifully, a large portion of a language changes very slowly. This is what allows present-day English speakers to read Charles Dickens and Thomas Jefferson unaided: The grammar and basic vocabulary are relatively similar to contemporary English. On the other hand, the lexicon has to change and adapt to meet our needs. This sort of change happens as new technologies are popularized (e.g. *walkman*, and the emergence of *click* as a transitive verb along with the appearance of the computer mouse), and as social, political or religious trends take hold and need terms for their definitive concepts, (e.g. *daycare*, *pollster*, *channeling*).

It is not possible to anticipate these changes to the language, so dictionaries and MT system lexicons must be constantly updated. In fact, lexicon management is the most basic aspect of the long-term maintenance required for a commercial product. Well-placed lexical expansion can go a long way toward improving performance.

It is worth mentioning that there are two other kinds of variation in text that lexicons cannot be updated to handle. The first is one-time lexical innovation: People often expect that translation of idiomatic phrases will be difficult. In fact, most idioms are frozen expressions such that it is not necessary to analyze them or derive a compositional meaning (made up from the sum of the parts) for the phrase or expression. The entire expression can be placed in the lexicon so that a culturally and linguistically appropriate equivalent is available. For example, (4) is a slightly archaic fixed phrase meaning ‘a good person to know’. It is always used as a noun, and would be easy to encode and translate.

(4) hail fellow well met

However, writers often playfully modify such expressions for effect. They can rely on the pattern-matching ability of readers to “get” associations with such well-known phrases. News headlines are full of this. For example, (5) appeared as a headline in *Newsweek* magazine that introduced an article about white-collar criminals networking in prison.

(5) Hail Felon Well Met.

When a phrase such as this cannot be matched as a unit, the system will attempt to analyze and translate each part, which will yield something meaningless or bizarre at best, misleading or offensive in the worst case.

The second kind of variation in text is grammatical errors and non-standard usage: One would expect that published, edited texts would conform to some sort of grammatical norm or standard. But in fact, people use non-standard expressions all the time. In addition, writers make slight errors that a human reader would never notice, but which a literal-minded parser cannot always adapt to successfully.

3.1.3 *The users' role*

One of the features that adds value to commercial MT systems is the inclusion of domain-specific dictionaries. The option to invoke a chemistry lexicon that contains domain-specific words and phrases when translating chemical texts can go a long way towards making a translation understandable and usable. However, there is a limit to how much such all-purpose subdomain lexicons can help, and users must be prepared to go the final distance in customizing the lexicon on their own (a quantified account of this problem is given in Section 4.2 below). For most commercial systems, this means assigning staff or hiring consultants or developers to do dictionary development. For data-driven systems the customization process can be automated if existing high-quality translations are available. In real texts, domain boundaries are not terribly clear: Vocabulary from one field is used literally or metaphorically in another, and many hybrid fields combine the concepts and terms of more than one domain. More importantly, many companies have their own lexicons of proprietary terms and terminological conventions used only at that company. For the same reason that such companies are likely to retain the services of a pool of dedicated translators who develop and refine their terminology lists over years, it will be necessary to augment the MT system's lexicon with proprietary terms in order to get satisfactory output. An MT development team can add extensive domain-specific terms to the lexicon, but because of specialization, proprietary terms, and the tendency for technical fields to overlap, it is unlikely that users will ever find a product that satisfies their needs exactly "right out of the box".

3.1.4 *Responsiveness and flexibility vs. focus and depth*

We mentioned above that development of an MT system is never complete. This disappointing fact stems from two causes: (a) Language is in a constant state of change. We described above the need for adaptation of the lexicon to domain or proprietary terminology, as well as for constant incorporation of new terms and evolving usage; (b) No theory or approach to date has adequately captured the full meaning of language the way it is actually used. The failure to account accurately for all aspects of natural language is a fact of life for MT developers; however, there are many ways in which developers can and do improve both the output quality of systems, and the usability of the tool itself. For example, they constantly monitor and evaluate each language pair to identify areas for improvement. These areas for improvement are used to define projects for linguistic development teams. New users of MT are quick to identify and point out errors in translation made by the system. They may approach the developer and ask, "Why don't you just fix this?" when they find obvious errors. Occasionally, development teams can fix a translation problem quickly and easily, but in general, newly discovered translation errors must be added to the "bug list", evaluated to determine the nature of the problem,

and then prioritized and scheduled for work along with all of the other problems waiting for attention. Within the framework of a rule-based system, modifications and enhancements need to be made carefully and deliberately, with adequate consideration of large amounts of data, lexicon tuning, and extensive testing. The debugging process must take into account not only the success or failure of the system in correcting the problem at hand, but must evaluate control data to ensure that improvements for one set of examples have not introduced degradations in the translation of other texts. This sort of development, while relatively time-intensive and slow, does pay off in consistently improved performance in the long term.

With finite resources, developers have to make choices about what to work on. High quality requires long-term focused effort. However a strategy focusing exclusively on such in-depth work would ignore market trends or sudden opportunities which require short-term projects to adapt systems to particular customer needs, or new text types, or implement new user features or tools in the interface. Management of a development group at an MT company requires careful prioritization of projects, and balancing of the opposing requirements for focused work and responsiveness to rapidly changing market demands.

4. MT for publication

4.1 Who's really using it?

Throughout its 50-year history MT has been viewed as either a threat to translators or a boon to those who need translation; sometimes as both at once. These notions are based on expectations that MT can produce translation at a publishable level without any sort of revision or post-editing.

The fact of the matter is that MT will never meet such expectations. Human translators generally do not produce publishable output after a single pass; most translation companies use several translators on a task to ensure that the results are correct. It is therefore totally unrealistic to assume a computer can do better. In fact, the reality is that, given the fact that computers work literally and linearly, it is apparent that they will generally do worse than a good professional translator will.

Does this then mean that MT is useless, as is held by many people, especially those in the professional translation community? In fact this view is equally false. Any number of corporate users can attest to the usefulness of MT as a *productivity* tool. The case studies published in *Language International*¹ are only a sample of the kinds of testimony MT garners in the commercial translation world.

The discussion here, then, will be focused chiefly on MT in a commercial setting where high-quality output is required for publication (the “dissemination”

application mentioned in the previous section). This is not to slight the ongoing use of MT as a means for gisting of web sites or documents. Such translation for assimilation is clearly useful, but users may tolerate poor quality and therefore skip revision altogether. For our purposes here, we will focus on the production of publishable output via MT.

Commercial use of MT is clearly governed by three key factors. First and most obvious is whether the language direction is available in a commercially robust system. As discussed above, not all language directions are available, for several reasons. Even some language directions that might seem significant are only poorly represented due to market demand. This situation will change in time as the commercial demands continue to rise.

A second factor is the suitability of the source text. Acceptability is a key issue here, since the quality of the output must be high enough to justify the work of post-editing. Whether MT is worth using with a particular kind of document is always measured against what it saves in the process. If the output from an MT system requires more work for post-editing than to do the translation from scratch, it is not suitable for use in the process. Over 50 years of MT research have made it clear that not all texts are translatable with MT at an acceptable level.

The most obvious documents which MT cannot handle are literary texts. One simply cannot imagine using MT to translate a Shakespearean sonnet or even the prose of Cervantes. The amount of work to post-edit it would be beyond anyone's patience. However, it turns out that other kinds of texts are also not particularly amenable to MT. General journalism is not cost-effective to post-edit (though unedited translations may be adequate for gisting), although financial reports and sports pieces may do well using a high-end system. Marketing materials are clearly beyond the pale when it comes to MT; there is far too much extra-linguistic content (e.g., culturally defined references) to be acceptable using a literal medium such as MT.

On the other hand, straightforward, generally unambiguous documents, such as technical documentation, help files, professional publications and general business correspondence can do quite well using MT. The case studies noted above represent corporations using MT with such documents. The results are quite good, as much as a 50% productivity gain over pure human translation.

Of particular commercial — as well as casual — interest these days is the use of MT for translation of web sites. This is a gray area for commercial MT since suitability hinges on the amount of work a translator saves in using such systems over working from scratch. Translation of a well-written informative web site is likely to be amenable to the use of MT in the process; a marketing-oriented web site is less likely for the reasons indicated above.

Finally, in addressing the issue of text suitability, we must look at another area of particular interest: automatic translation of e-mails. On one hand, this is really under the translation-for-communication rubric. As a relatively new application of MT, much less is understood about how it is being used and how effectively MT is applied. On the other, however, there is commercial interest in the use of MT for restricted e-mail communication within a corporate setting, e.g., between engineers in laboratories on either side of the Pacific. This has real possibilities for MT as long as the style of the messages falls within the limitations of the MT system.

In considering the limitations of text suitability, then, it may be stated that some texts are inherently suitable while others simply are not. If the text is judged suitable, MT can contribute significantly to translation productivity gain.

The third factor is dictionary coverage. Note the issue here is not size, but coverage. A large general system dictionary may not be useful if it does not contain the particular terms with their appropriate translations needed for a given text or text corpus. Typically, corporations have anywhere from 10,000 to 70,000 terms that are used with their products. A typical document uses some subset of this lexicon. Interestingly, the terminology overlap, even between companies in the same industry, may be as little as 20%. This means that the MT system's dictionary must be tailored for a particular corporate terminology database and for particular documents. The point is not to cover the entire vocabulary, but what is needed for specific texts. Once done the productivity gains can be significant. Further, MT systems provide an excellent means for terminology management and control, an area of rising interest in commercial settings.

In summary the three factors that are key to the success of MT for commercial translation are:

- Availability of the language direction
- Suitability of the text
- Dictionary coverage

If these three factors are met, the chances for success in using MT as a productivity tool are high. The experiences of successful commercial applications of MT underscore these views.

4.2 How to succeed with commercial MT

Given the three factors, MT would seem to be a very limited tool, at least for publication-quality translation. The reality is that success is often achievable—but at a price. Commercial MT applications require a significant up-front investment on the part of the user in customizing the system or paying to have it done. Additionally the user must be willing to maintain the system over time as terminol-

ogy evolves. Data-driven translation systems promise to automate this process, for example with automatic terminology extraction, and automatic learning from existing translations. However, the terms and translation rules learned will only be as good as the human translations they were based on. Texts and translations that contain inconsistent use or translation of terminology will not yield consistent, high-quality translation rules. Terminology management is as key to high-quality, consistent machine translations as it is to human translation. The pay-back is proportional to the investment. Viewed as a productivity tool for use by the professional translation staff, an MT system will pay for itself in a very short period of time, but the return on investment requires the investment. If there is no desire to make the investment or to maintain the system, it is best not to venture into the enterprise. Probably the most significant factor in this is the buy-in of all those involved with the system from the translators to upper management.

The initial investment and the maintenance of the system demands a level of support from the MT system vendor beyond that given by most software sellers. MT is not now a plug-and-play application on the commercial level. Tailoring the dictionaries to get the necessary coverage and assessing translatability often requires backing from the vendor's customer-support staff. Further, the translation process within the corporation may have to be changed. Using MT is not the same as using human translation. The issues are often more linguistic than technical. Extensive training is essential, followed by working with the vendor's help desk on a regular basis. Some corporations have established controlled (or restricted) language and/or authoring standards as a means to get more out of MT systems.

In evaluating commercial MT systems, the potential customer must take the level of available support and training into account. If a vendor cannot give the client the necessary guidance and service, the implementation of the MT system is likely to fail. Linguistic support must be available in addition to technical. If the potential client is unwilling or unable to do the dictionary work, some outside source must be found to get the terminology into the appropriate database.

The fact that a number of corporations have been successful with MT indicates that it can be done. What must be in place is the corporate willingness to invest in the process and the necessary support and training from the vendor.

4.3 MT and translators — you can't hurry love

MT, as presented here, is seen entirely as a production tool for professional translators in a commercial setting. The question arises, however, why the translation community did not embrace the technology long ago. The answers are many and complicated. We will address the prevalent ones without attempting to exhaust the issue.

First and foremost is the fact that MT for much of its history has been presented as a means to eliminate translators. This was clearly the early view in spite of Bar-Hillel's (1960) warnings; unfortunately the lessons of MT history have often been lost in the desire for cheap and easy solutions to translation needs. Given their knowledge of the complexities of translation and the standards of the profession, most professional translators have viewed MT more as a threat than an aid. The mistaken notion that MT will do the whole job has resulted in the translation community dismissing it as doing nothing.

A second answer is that the translators of the world have fine-tuned their craft over centuries and are generally suspicious of any technology that claims it will change their work virtually overnight. In the beginning of the 21st Century some translation is still done using techniques which were in place 50 years ago or more. Introduction of computer technology of any kind into a well-established process is potentially traumatic and disruptive.

Third, MT often did not live up to its own expectations. Promises of the past were often not fulfilled. This situation has now changed. Corporations who work with MT vendors have a great deal of influence on the R&D efforts. Systems are now commercially viable in ways they were not before.

Finally the three factors inherent in MT make it difficult to use outside of the corporate setting. Most independent translators and small-to-medium translation companies do not have the resources to build dictionaries or pick and choose texts for MT. Most freelancers and smaller companies do work for a large variety of clients over different domains, precluding the development of the necessary dictionary coverage. Further, there is little possibility of investment in a system. This has been a very real barrier to the adoption of MT in the commercial market of independent translators and small to medium translation companies. Given the fact that most corporations rely on the freelancers or such companies for translation, the market for MT has been limited.

Clearly these factors are changing. Translators are coming to realize that technology, including MT, can be a boon to them. The translation community is adopting a more practical, less reactive approach to the limitations and potential of MT, even as the demand for commercial translation is growing at what is predicted to be 30% per year. Pressures to produce more and more translation with less and less time are driving professional translators to seek solutions in all quarters including MT. In addition, the Internet is offering a means for freelancers and small-to-medium companies to use the technology without the major initial investment required for MT in the past.

The future, then, appears to be bright for MT as a tool in the commercial translation process. In fact, it is clearly coming into its own just when the demand for commercial translation needs it.

Notes

- * At the time of writing of this chapter, Scott Bennett was working for Logos. He previously also worked on the *Metal* system, later commercialised as *T1*. Laurie Gerber worked for several years for Systran Inc.
- 1. See for example Cremers (1997), Lange (1998) and Marten (1998) and other case studies reported in *Language International*.

References

- Bar-Hillel, Yehoshua (1960) "The Present State of Automatic Translation of Languages", *Advances in Computers* 1, 91–163.
- Cremers, Lou (1997) "Implementing MT at Océ", *Language International* 9.6, 16–17.
- Knight, Kevin (1997) "Automating Knowledge Acquisition for Machine Translation", *AI Magazine* 18.4
- Lange, Carmen Andrés (1998) "Tying the Knot", *Language International* 10.5, 34–36.
- Marten, Laura (1998) "Machine Translation Finds a Home at Mitel", *Language International* 10.3, 38–41.
- Yang, Jin and Laurie Gerber (1996) "SYSTRAN Chinese-English MT System", *Proceedings of the International Conference on Chinese Computing '96*, Singapore, June 4–7, 1996.

CHAPTER 12

Going live on the internet*

Jin Yang and Elke Lange
Systran Software, Inc., La Jolla, CA

1. Introduction

With the goal of “eliminating the language barrier on the Web” (AltaVista, 1997), AltaVista teamed up with Systran Software Inc. to offer the first free-of-charge online translation service *AltaVista Translation with Systran*.¹ Global accessibility, intuitive ease of use, and near-instantaneous real-time translation speed were teamed up with *Systran*’s proven MT technology. Ten major European language pairs were offered in the initial phase, translating English to and from French, German, Spanish, Italian and Portuguese.

The translation site’s domain name, *babelfish*, is a concept taken from the book *The Hitchhiker’s Guide to the Galaxy* by science-fiction author Douglas Adams.² In the book, galactic hitchhikers had an easy way to understand any language they came across: simply popping a “small, yellow, and leechlike fish” (a babelfish) into their ears. Similarly, the translation service aims to point the way toward the future of a global Internet, giving increased access and understanding to millions of multilingual documents. Today, English is still the dominant language on the Internet (just under 50%), closely followed by the other major western European languages, but more and more documents are becoming available in a greater variety of languages. Also, the user base is rapidly changing away from that consisting mostly of English speakers.

The service is available directly from AltaVista’s Search Service. As shown in Figure 1, The user can input text or a URL of a web page in the box, choose the language direction, and click on the *Translate* button, and the translation comes back instantly. The translation service is also accessible from a search result, as seen in Figure 2. A *Translate* link is present below each search result.

By clicking on the button, users go to the *babelfish* page. Users are also encouraged to link to the site as much as they like. Many web sites offer live online translation of their web pages via the *babelfish* translation service. For example,

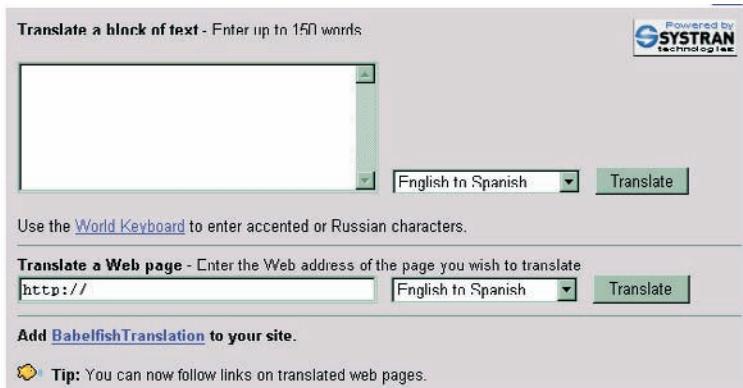


Figure 1. *Babelfish* front page as it appeared in November 2002.

4. History of the Internet
 History of the Internet. Unless otherwise noted, the sites listed in this directory are provided by Congress...
URL: lcweb.loc.gov/global/internet/history.html
 Last modified on: 17 Dec 1998 - 10K bytes - in English
 [Translate] [More pages from this site]

5. History of the Internet
 Slide 1 of 36. History of the Internet. Paul B. Barron. AS 214 - Leadership and Management
URL: sun.vmi.edu/hall/as200/as213214/WEBHIST/sld001.htm
 Last modified on: 15-Mar-1999 - 10K bytes - in English
 [Translate] [More pages from this site] [Company factsheet]

Figure 2. Search results including “Translate” button.

viola.com has (or used to have) translation links to *babelfish* as shown in Figure 3 in their home page.

Systran had pursued online translation before this service. The first implementation was in France, where *Systran* translation systems have been used since 1988 on Minitel, an online service offered by the French Postal Service and widely available to the public on dedicated terminals. Whereas initial usage was by curiosity seekers, translating e-mail and simply experimenting with translation, later usage shifted to more serious translation of mostly business correspondence. The drawback of this service is that it is expensive, relatively slow, and not easily integrated with the PC environment. Since early 1996, Systran has been offering online translation on the Internet via a service geared mostly toward web-page translation. *Systran* is also used on two major intranets: the first within US Government agencies, and the second in Europe, where *Systran* has been the official MT system of the European Commission (EC) since 1976. Currently 1,800 professional



Figure 3. Translation button included in web page.

translators access the EC systems on their internal network, via e-mail.

The AltaVista translation with *Systran* has pushed online translation a big step forward, with a good implementation realizing the primary requirements of online translation: speed, robustness and coverage (Flanagan, 1996). Also, accessibility pushes it to the forefront of worldwide awareness. Being one of the most trafficked web sites, AltaVista's site makes the service accessible to all, and it is very easy to use. Powerful DEC Alpha servers and the fast AltaVista Search Network complement *Systran*'s high-speed translation turnover. *Systran*'s time-tested MT technology provides good quality of translation with a wide coverage using broad and specialized dictionaries and linguistic rules. This truly remarkable combination made the real-time online translation a tremendous success story.

The translation page has been acknowledged as a good web site by various sites (e.g., *What's Cool: Netscape Guide by Yahoo!*). About 14,000 web sites already have been found to contain a direct link to the translation page, which helps to generate translation traffic. The media's reaction is explosive, with comments and introduction to the service in particular and MT technology in general (published in assorted newspapers).³ The public's reaction is also overwhelming. Our *babelfish* association with AltaVista is now known well enough that people recognize it by a nickname, as illustrated by this quote from an enthusiast's news page of September 1999: "... the article are [sic] in French — so head over to the Fish and try to make heads or tails of the translation".

The *babelfish* translation service has been available to the general public for over six years. During this time, usage has increased steadily from approximately 500,000 translations per day in 1998 to approximately 1.3 million per day in 2000.⁴

2. Real-time translation on the internet

Translation is a difficult human skill that, in itself, is often underestimated and that to date has not been duplicated by any software. Yet, expectations by the general public are usually very high when it comes to judging MT output. So, how does one prepare an MT system for coming out on the Internet? In particular, which special issues need to be addressed for a system like *Systran* that has been in use in many different ways over many years?

First, there are technical challenges to address and then there are linguistic areas that suddenly become more important than they ever had been.

2.1 Engineering requirements

The translation service uses the *Systran* translation servers running on three DEC Alpha 4100s, with 1gB RAM and three 500 Mhz CPUs. The Web Server is Apache, and the translation CGI is written in the C language.

The configuration is presented in Figure 4.

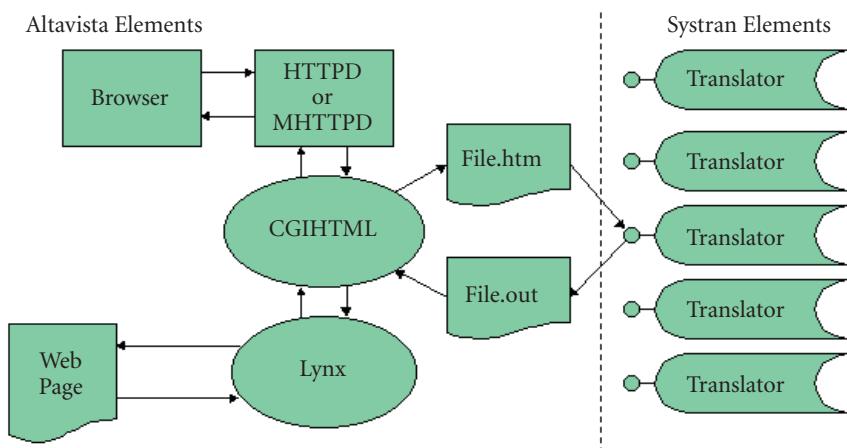


Figure 4. Technical configuration of *babelfish* service (Story, 1998).

For an MT server, the basic requirements for providing on-line translation on the Internet includes robustness, format handling and speed.

Translating web pages requires the system to handle HTML tags — the “markup” language used to determine what web pages look like — in order to preserve the formatting of the translation. This process is called “format filtering” in *Systran*’s programs, which include the pre-filter and post-filter stages. The pre-filter distinguishes HTML tags from the text and understands the meaning of the HTML tags. For example, in (1), the tags in angle brackets indicate the start and end of a level-2 header, which will probably appear in large bold characters.

- (1) <h2>This is a header</h2>

Other HTML tags indicate formatting, tables, background and text colors, links to other pages, and so on. The post-filter replaces the tags around the translation in a new HTML page, which preserves all the tag information from the original file, but with translated texts.

This implies that *Systran* must understand the HTML tags in order to pass proper sentence-end information to the translation engine, because the translation unit of the *Systran* translation engine is a sentence. The web pages usually contain many stand-alone noun phrases in lists and tables. In many web pages, a line-break
 is used to indicate the end of a sentence, instead of the paragraph-break symbol <P>.

Font information is usually used around an entire word or a group of words, for example (2a). Difficulties arise with examples like (2b), where only the first letter is bold. The translation engine has to ignore the font information, but reinstate it afterwards, for example giving the German (2c). Especially important is not to translate words recognized within URLs.

- (2) a. Agile
- b. Agile (appearing as Agile)
- c. Flink

2.2 Linguistic requirements

There are many linguistic requirements for web-based translation above and beyond the normal requirements for MT. The example in (2) points to one such case, where the bold initial letters might have had a special significance, e.g. spelling out an acrostic, which would impinge on the translation. Another case is exemplified in (3), where the English acronym happens to coincide with a word, resulting in the comic mistranslation as shown.⁵

(3) WHO (World Health Organization) *WER* (*Weltgesundheitsorganisation*)

Web pages often contain proper names, most of which should not be translated (though some should, for example *London* is *Londres* in French, though not when referring to London, Ontario), and titles which present a similar problem (especially tricky are film titles, which are sometimes rendered quite opaquely from one language to another: the Beatles' film *A Hard Days Night* is called *Quatre garçons dans le vent* in French.⁶

More problematic is the idiosyncratic spelling, punctuation, capitalization and, in the case of languages other than English, misplacement or omission of accents. As has been mentioned in other chapters in this collection, MT is very sensitive to style and technical domain, which are very varied in web pages. All of these factors contribute to a great variety in the quality of the translations offered by *babelfish*.

3. User feedback

User feedback is encouraged via a panel (Figure 5) in the *babelfish* web-page. Concurrent with the increase in usage, user feedback also has increased every month. Between January and May 1998, 5,005 e-mails were received concerning linguistic and/or translation comments alone. This is the set of user feedback discussed in this paper.



Figure 5. Feedback panel in *babelfish* web-page.

3.1 User reaction

Most users are enthusiastic about the service. People say they never imagined that something like this exists. Many who have never used translation software and never considered purchasing one are now trying it out.

This is Very Cool!!!! Fantastic! Fantastique! Fantastisch! Fantastico! What else can I say?

And all I can think of is “wow.” I know some foreign students this will help tremendously! And I will certainly find it useful in future correspondence with contacts around the globe.

I actually do not know how long you have been offering this service but it is an absolute success! Congratulations on an excellent service which is not only very accurate (I speak several languages myself) but nice and fast as well!! This is the best initiative that I have found on the Internet so far. Keep up the excellent service!

Another group, especially professional translators, sent angry e-mail to protest against the initiative.

Your “translation program” on the Net is a worthless embarrassment. It is good for nothing, except perhaps for a cheap laugh... I am a better judge of translations than you are. You should hang your heads in shame.

Sorry, no gushing praise. The translation was incomprehensible; half the Portuguese words were not even recognized. Back to the drawing board.

Some are confused, wondering whether the translation was done by a human or computer.

I would be very interested to know if the entire translation is via software or if natives or language trained staff review the work.

You should fire your translators and hire me.

Also, similarly to the experience reported by CompuServe (see Flanagan, 1996), many professional translators seize the opportunity to offer their services by sending resumes.

May I suggest, in order to improve the output quality and accuracy of your MAT-based translations, that you maintain the software but add a freelance team of experienced translators, such as myself, as post-output editors?

I have just discovered your service and am very excited about it since I am a professional translator and sometimes need some help. I asked for several terms of which two very important ones were wrongly translated. Those are “mainstreaming” and “pork barrel”. English is full of those words or expressions that are hard to translate. It would be great if you could really find accurate translations for

those terms which, in the first place, are usually created in the American environment and are sometimes very difficult to translate.

As a professional translator, I suggest that you seriously revise your English to French translation. This cannot be used as a reliable tool for people who do not master the foreign language translated.

This might be used to translate very simple sentences however complex sentences should be revised by a professional translator. While trying out the system, I came across many grammatical errors and missing words making the sentence illogical. If you are looking to improve your software and need someone to test it, I would be more than happy to assist you!

I entered two quite simple sentences (English-German). I am always glad to see I won't be disposed of in the near future. Both sentences were not understandable in the translation. I think the problem is always multiple meanings of words.

Being a professional translator myself, I am always intrigued with machine translation, but I'm sure it won't work for a long time yet.

PLEASE, PLEASE, REMOVE THAT AWFUL TRANSLATION FROM THE NET IMMEDIATELY. IT MAKES ABSOLUTELY NO SENSE AND CAN ONLY PROVIDE A SPANISH SPEAKER WITH A REASON TO LAUGH AT YOUR ATTEMPT. I HAPPEN TO BE PROFESSIONAL TRANSLATOR AND CAN TELL YOU THAT THE COMPUTER CAN NOT REPLACE US IN THESE ENDEAVORS. PLEASE CONTACT A PROFESSIONAL TO REDO SUCH GIBERISH TO A CONCISE, LOGICAL STATEMENT.

In summary, user feedback consists of approximately 95% praise, sprinkled with friendly bug reports and suggestions. Less than 5% of users disparage the service.

3.2 Acceptance of MT

CompuServe's two-year on-line experience reported that users were first amazed, then disappointed, and finally pragmatic about the quality of translation (Flanagan, 1996). In our experience, we found that the majority of users are amazingly understanding of MT capabilities and limitations. They are impressed by the capability and even the translation quality.

Some works better than [sic] others. But all told, this stuff is amazing. In the blink of an eye I got most of the gist of something from Italian. Technology can sometimes be breathtaking. This was one of those times for me.

All in all, though, I was impressed at generally good accuracy, keeping the phrases to be translated simple, of course, and the speed: less than 30 sec. at approx. 50kbps.

Many users show that they know what to expect from MT: less than perfect results. They are willing to help.

Generally, I'm impressed. Congratulations! Of course not perfect. But — who knows what you can do in 2 years (or you have already done and not yet disclosed ☺). I would like to support you by sending you these little bugs. ...

I could provide you with some software-related phrases and terminology extracted from Italian software source code comments, if it would help you folks to do a better job of translating it to English.

Users also realize the challenges of MT: name handling, idiomatic expressions and context-sensitive translations. Some of them even “point out” ways to future success.

Pretty good translations there — I'm impressed. You need some help with the idioms!

I think the context problem would be very difficult to solve but what about a certain idiom library of often used terms?

SUGGESTION: To design a translation mechanism that is grammatically accurate, and accurate in context. Many expressions cannot be translated literally from one language to the next, so I suggest that more careful consideration is given to idiosyncracies [*sic*] and nuances in translations. This will allow you to provide more accurate translations and better service

The positive feedback shows that MT has been accepted as a useful tool in the online environment. It is gaining worldwide popularity, with the “not perfect” quality.

3.3 User evaluation

How does the user judge the translation quality and the usefulness of the service? Regarding translation **quality**, many users take simple every day words, and check the translation, for example *cheese, mumble*. This use of the service is so widespread (more than 50% of translations are of one- or two-word phrases) that we are considering adding a button to the web page to distinguish dictionary look-up from translation. One-word translation requests would be treated as dictionary look-up, and a list of alternatives, perhaps with glosses, will be returned.

Many users try to trick the system with idioms, as in (4). This is a regular feature also of negative reviews appearing in newspapers. It is an easy way to elicit a bad translation, but really does not tell the user anything about the likely overall quality of translation (any more than if it gets the idiom right — because it happens to be in the lexicon — is a sign of a good system).

- (4) a. It's raining cats and dogs.
b. All is fair in love and war.

The following comment illustrates the linguistic naïveté of users, who apparently expects *may* to be translated the same as *May*.⁷

From English to whatever: may (by which I mean the month). The most stupid piece of translating software I've ever seen... I just discovered that IT is case sensible, concerning months. Amazing.

A method very frequently used by users to evaluate the translations is **round-trip** (or “back-and-forth”) translation. If the user does not know the target language, they judge translation by translating the results back into the original language (also used as a source of entertainment, see below). This seems to be a very widespread and intuitive thing to do, though MT researchers generally warn against it, and we have considered including a specific warning of its “dangers” on the web site, or in the list of frequently asked questions (FAQs). The problem is of course that even a small translation error in the first place produces a bad source text for the “return journey”, so the result is like the product of the errors (not just the sum). On the other hand, an excellent return translation equally tells you nothing about the translation quality: it is most likely that the phrase or sentence can be translated fairly literally in one direction and, it follows, fairly literally back again. Back-and-forth translation would be fairly easy to detect automatically, and we have considered incorporating software to output the original source text in such cases!

Evaluating the **usefulness** of the output is much more difficult. Given the state of the art of MT, we know that unedited translations of unrestricted text may be of low quality, and therefore should only be used for information-gathering purposes. On the other hand, carefully written texts may produce translations that are near publishable quality. Unfortunately, *babelfish* cannot guarantee the quality of the output: at best we can only offer a guide to users to get the best out of the system. Our FAQ page contains the following advice:

What Documents and Text Translate Best?

How to Use Babel Fish contains examples of newspaper sites you can try. Newspapers are typically well written, use proper grammar, and translate well. When you write for automatic translation, use short sentences and avoid slang, idiomatic expressions, and unnecessary synonyms.

3.4 How is *babelfish* used?

The usual expectation is that MT, in the online environment, acts mostly as an assimilation tool. Our experience shows that the use of MT is going beyond that. We have identified five functions for the online translation service.

3.4.1 As an assimilation tool

Information assimilation is the primary purpose of translating web pages. Users find it useful to get the information they want. They do not worry too much about the fine points of the translation, especially if the translation gives a good sense of a foreign-language newspaper article or other piece of information.

It does not matter one whit that language translation is not 100%, nor even 90% accurate — getting the “gist” of a foreign-language webpage (and fast!) matches the impedance of web attention spans.

3.4.2 As a dissemination tool

People translating their own web pages hope for greater dissemination of their message. Some users put a link to the translation service page. With a simple click, a personal or business web page can be translated into other language on the fly. This is a sensitive area, since the imperfections of MT may distort the message. Suggestions were made to mark such translations with a warning that it was MT output.

3.4.3 As a communication tool

Most users are happy to be able to communicate in a language they do not know, and they accept MT as long as the message conveys the idea of what they want to say.

Your software has enabled us to give a much needed job to a woman in our neighborhood (now our housekeeper) who speaks only Spanish. We are able to leave her instructions regularly and she may now ask us any questions she might have. No it isn’t perfect, but darn close.

This is the best. I can finally write my grandmother. She doesn’t speak English, and I don’t speak Portuguese. This is enough to make me cry. Thank you very much. The translation facility with AltaVista is terrific. It would be great if [sic] somebody could build a chat room with built in translation. I would love to converse with somebody who does not speak English. This could be a fun way to learn another language.

3.4.4 As an entertainment tool

One of the most popular usages is back-and-forth translation, as mentioned above. Jeff Mode, writing in the on-line magazine *ZDNet* in August 1998 wrote:

The inexactitude of machine translation becomes especially noticeable when a fragment of text is converted from one language to another and then back again, or through several languages, the ‘drift’ increasing with each pass of the software.

As MT developers, we would like to discourage this practice, especially when users

attempt to judge translation quality by evaluations of the back-translation. However, it has become a very popular **entertainment**. Even the famous Italian author and poet Umberto Eco could not resist the temptation to play word games with MT.⁸

I did English/Spanish and English/French and back-translated. All samples came out entirely understandable.

Your translation from English to Spanish and Spanish to English may not be working well. Because I tried, what is Y2K? I translated to Spanish. From Spanish to English, it shows which is Y2K?. I do not know translation to Spanish or Spanish to English correct or not.

menu (English) → menu (French) → finely (English)

I noticed that the Spanish-to-English seems to work better than English-to-Spanish, is that because some guesswork is going on? Basically, I noticed that the English words that resulted from my first Spanish-to-English translation did not translate back into the same Spanish when I used English-to-Spanish.

One person set up a “para-site”, now disappeared, called *The AltaVista Language Transmogrifier (sic)* to take advantage of this, allowing surfers to run passages of text through up to five languages with a click. Another web site, also now disappeared, sought to stress-test the system. Various round trips (e.g., multiple round trip through a single language, serial round trip through five languages, etc.) were tried to translate the English idiom *get with child* (i.e., ‘impregnate’) back and forth. This kind of process is also shared in the user feedback. Poems, jokes, and idioms are often tried in this process. One web site contained a greeting card with the note “Translation courtesy of Alta Vista and Systran”: It consisted of multiple round-trip translations of *Happy New Year*. As our FAQ page states,

Translating languages is a very complex task. The translator works best when the text you wish to translate uses proper grammar. Slang, misspelled words, poorly placed punctuation, etc. can all cause a page to be translated incorrectly. Also the more translations a piece of text goes through, the further the resulting meaning will be from the original. So you will not necessarily get a good idea of the quality of a translation by translating the translated text. But you will have a lot of fun.

3.4.5 As a learning tool

MT was never meant to be used to teach languages (see Chapter 17); however, there seem to be users who hope to use it as a learning tool. Amazingly, students are using it to do their foreign language homework.

LOVE your site — had a lot of fun testing my French! I imagine this will be a great success with the international business community — not too [sic] mention students “helping” themselves with their homework! Wish I’d had it in high school!!!

The only thing that would make it better is if you have an audio where someone could speak the translation so that I would know the proper pronunciation.
I find your service invaluable in untangling Italian verb forms, moods, tenses, etc.

4. Text analysis

We have attempted to study some characteristics of the documents translated using the service. We randomly picked two days' worth of translation log files from AltaVista: one is Monday, June 22, 1998 and the other is Wednesday, November 10, 1999. Table 1 shows the total number of translations on those two days.

Table 1. Total number of translations on two census days

Date	Total number of translations
1998–06–22	370,990
1999–11–10	740,218

Babelfish offers both text and web-page translation. At the beginning of the service, the translation of each accounted for 50%. Then the proportion moved to 40% (web page) vs. 60% (text). This trend continues. The latest statistics show that about 80% of the translation requests are text translation, while only 20% are web-page translations. Table 2 gives a detailed account of the two-day translation log.

Table 2. Translation type (Text vs. Web-page)

Date	Total	Text	Web page
1998–06–22	370,990	214,051	57.6%
1999–11–10	740,218	609,800	82.4%

Among the ten language pairs, English–Spanish and English–French have been consistently the most two popular languages, as Figure 6 shows.

The text translation allows the user to type a limited number of words in the translation box. A detailed check of the numbers of words in each translation shows that about 50% of translations have less than five words. The average is around 20. Table 3 gives as an example figures for Spanish texts. Data for other languages are similar.

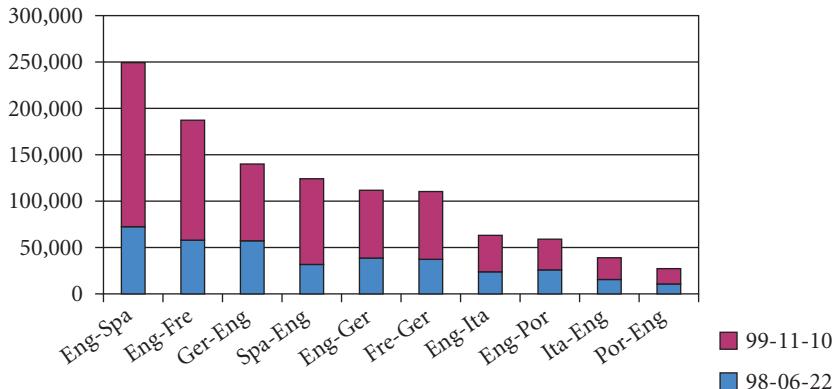


Figure 6. Distribution of language pairs

Table 3. Length of texts submitted for translation

Number of words	98-06-20	99-11-10
1	22.6%	26.3%
2–5	25.5%	26.6%
6–10	13.7%	14.3%
11–20	10.7%	9.9%
20–100	12.2%	9.8%
100–...	15.2%	13.2%
Average	24.17	21.12
Longest	212	266

Of some interest is the relative frequency of individual words. Obviously, function words such as *of*, *the* and so on figure highly. Perhaps more surprising and indicative of how the users use the service is the following list of words which all figure significantly highly in the list: *time, love, know, want, good, new, see, use, information*. A number of phrasal collocations are quite frequent, among them *make love* and *make progress*.

One observation we can make is regarding the high volume of input which is “badly” written, and thus results in mistranslations. We already saw the example of the user expecting *may* (with lower case initial) to be recognized as the name of the month. There were many cases of bad spelling (e.g. *bad-tempered*, *basicly*, *Saterday*), unhelpful or missing punctuation and capitalization (especially the first-person pronoun written as *i*). Bizarrely, some users would type in complete garbage, such as (5) — one wonders what the user really expected in this case.

(5) of of of of of

A lot of the input was very informal in nature. Greetings and conversational phrases figure prominently, such as *Hello*, *How are U?* (*sic*) and so on. As one might expect, Internet jargon is widely found, including e-mail abbreviations like *IMHO* ('in my humble opinion'), *M2cW* ('my 2 cents' worth'), *FYI* ('for your information'), and the ubiquitous "smileys". Chat-room jargon is particularly difficult because of the tendency to abbreviate content words to speed up typing.

The internet contains X-rated material. This area of the activity is also reflected in the type of text translated. The word *fuck* is among the top 300 high-frequency words (including function words). In a random choice of 200 input texts from our sample, we found about 10% of texts were sex-oriented in content. In another survey of the material we looked at all verb-object collocations and found that 5 of the top 50 verb-object pairings were sexual in nature.

Among the broad usage of the translation service, translation of such expressions is quite prominent. With current sensitivity to X-rated material on the Internet, we reacted to a mother's complaint when her child translated harmless text and got translations with sexual connotations. The specific example was not given. In fact, some of our dictionaries contain a number of risqué terms, entered during the early days of Minitel usage in France. When we set a switch to hide these terms for the sake of concerned parents, a number of other users complained that the system couldn't handle the "adult" material. Although it is not the job of MT systems to censor this kind of material, translating them is certainly not the system's greatest expertise.

5. Discussion

5.1 Legal implications

Questions have been brought up regarding the legal implications of MT online, though no answers are provided (see Westfall, 1996). We are not going to answer the question of existing and coming legal implications either, but can share some experiences.

Some users point out the need for a clear disclaimer in the online environment. Creation of a detailed and standard disclaimer is a worthwhile task for the MT industry as a whole.

I suggest you attach a detailed disclaimer to this service and seek ways through which the results could be made more flexible and accurate.

One of the frequently asked questions is "Can I trust the translation?". The answer provided in our FAQ web-page is as follows:

Machine translation produces reasonable results in many cases. But you should not rely on it. If you want to send a translated text to another person or use it in correspondence, always explain that you are using an automatic translator named Babel Fish and append or reference the original text. This acknowledgement will put the translation into the right context and will help you avoid embarrassing misunderstandings. When it's important to have an accurate translation, ask a human translator to polish the Babel Fish translation.

Copyright questions have not yet come to our attention. However, we had one user who was ready to sue us when he saw the translation of his web page and found the name of his company translated and mangled. Fortunately a quick correction of the problem could be made.

5.2 Possibilities and challenges

Our experiences have given us an opportunity to look at translation quality in a new light. Automatic identification of language, domain, and style level are needed for a translation service catering to such a wide audience. While such parameters can be specified in the regular versions of *Systran*, the AltaVista service does not offer such choices for reasons of ease of use. Automatic identification of these parameters would be one more step toward enhanced user friendliness (see Lange and Yang, 1999).

Further increasing speed and efficiency, and lifting translation time and size limitations are other items important for the fast turnover of large volume translation.

Users are a valuable resource for the MT developer. Their specific bug reports and general suggestions can be catalogued and acted upon as an important step toward the goal of enhanced quality and coverage. Channeling user input, therefore, is added to the list of tasks for the MT developer.

5.3 Conclusion

The AltaVista translation service with *Systran* is a good showcase for MT technology. The explosive and positive user feedback shows that MT has proven its worth in practice. Improving translation quality and expanding language coverage are definitely pressing challenges. MT needs to earn its keep, and the best way is through more good implementations.

6. Postscript: Chatting multilingually

In a development no doubt encouraged by the success of AltaVista's *babelfish*, some on-line chat-rooms also offer a multilingual capability.⁹ A good example of this is

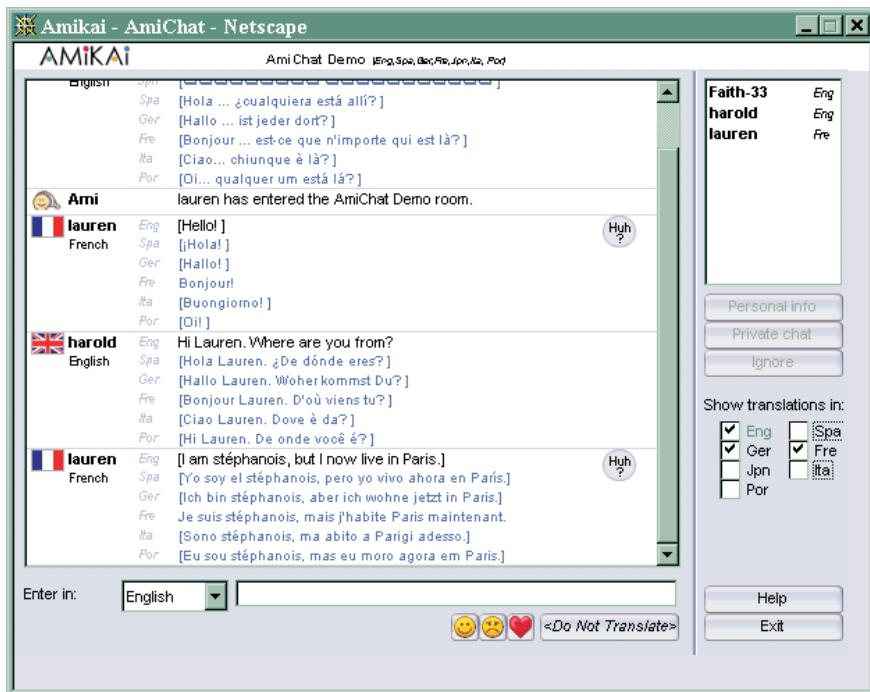


Figure 7. Screen capture of multilingual chat hosted by Amikai.com.

Amikai's "AmiChat" facility at www.amikai.com (they also provide web-page, e-mail and text translation). Users can chat online in any one of eight languages (English, French, German, Spanish, Italian, Portuguese, Korean and Japanese) and see their contribution translated into any of the other seven. Figures 7 and 8 show an example of a brief exchange between Harold, typing in English, and Lauren writing in French.

Harold, who is a computational linguist, is showing on his screen (Figure 7) output in six of the languages. Lauren on the other hand is only interested in French and English (Figure 8). The system has a nice feature: the "Huh?" button which, when pressed, generates a canned message suggesting that the text is badly translated, and asking the chatter to rephrase it. We can see this in Figure 8 where Harold's comment, *I'm not sure what "stephanois" means*, is mistranslated. The system is quite robust, for example when Harold types in *Allez les verts*, despite the fact he is supposed to be typing in English. It is not robust enough however to cope with the football (soccer) terminology *at home* (which should be *à domicile* in French), and the colloquial abbreviation *Man U* for *Manchester United*. Nevertheless, it is clear that a multilingual dialogue can be facilitated in this way.

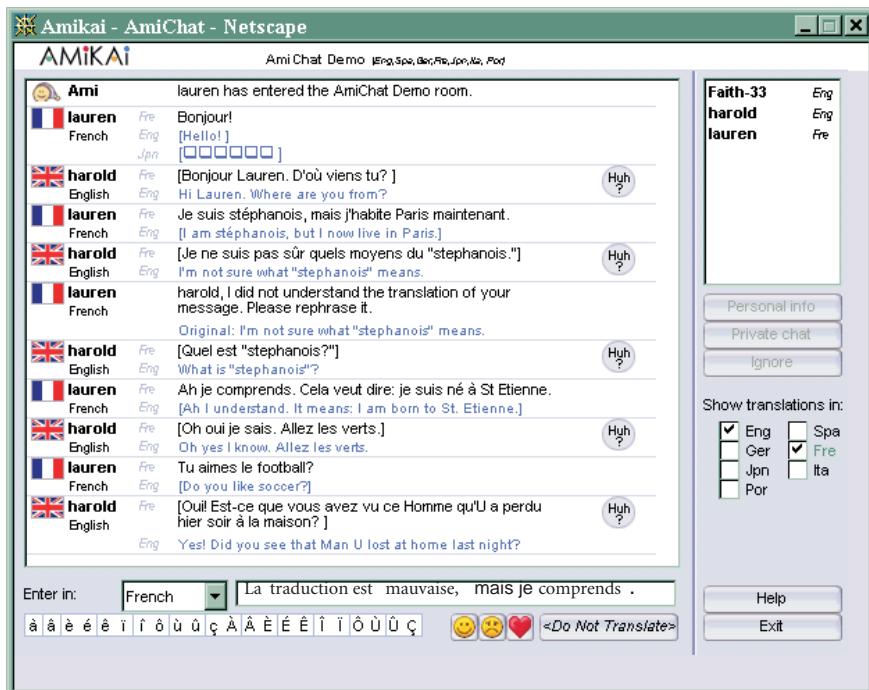


Figure 8. The same chat as seen from another perspective.

Further reading

This chapter is developed from an earlier version which appeared as Yang and Lange (1998). In addition to the sources cited in this chapter, the following are of interest: Bennett (1996) discusses users feedback, Choi et al. (1999) describe a Korean web-page translation system, Gerber (1997) and Gerber and Yang (1997) discuss commercial development of *Systran*, Miyazawa et al. (1999) evaluate web-based MT services, Nakayama and Kumano (1999) discuss the use of the web to accumulate dictionary data. Amikai's chat-room is described by Flournoy and Callison-Burch (2000).

Notes

* We would like to thank Dimitris Sabatakakis of Systran for his comments and suggestions. We would also like to thank Henry Story of AltaVista for valuable discussions. In this

(and other) chapter(s) we distinguish typographically Systran Software Inc., the company, and *Systran*, its MT system.

1. The URL is <http://babelfish.altavista.com>.
2. Douglas Adams, *The Hitchhiker's Guide to the Galaxy*, Pan Books, London, 1979.
3. The following examples are from newspapers and journals published in the USA in the early part of 1998: Kevin Maney, "Translating via Web is a Hoot as well as a Help", *USA Today*, January 22, 1998; Bruno Giussani, "Free Translation of Language Proves More Divertimento than a Keg of Monkeys", *The New York Times*, March 10, 1998; Laurent Belsie, "Translation Software Rides Roughshod over Idiomatic Speech", *The Christian Science Monitor*, March 19, 1998; Kurt Ament, "Real-time Machine Translation on the Internet" *Intercom*, May, 1998; Tina Kelly, "Even Helpful Translation Software Sometimes Weaves a Tangled Web", *The New York Times*, April 30, 1998.
4. Stated by Henry Story during Panel Session at Third Conference of the Association for Machine Translation in the Americas, Cuernavaca, Mexico, October 2000.
5. The German word *wer* corresponds to the English pronoun *who*.
6. The title literally means 'Four boys in the wind', though *dans le vent* also has an idiomatic meaning 'up to date'.
7. The use of the word *sensible* rather than *sensitive* suggests that the user may be French, and may not remember that month names in English are written with a capital letter, unlike in French.
8. As reported by Kevin Maney, see footnote 3.
9. This section has been added by the editor.

References

- AltaVista Search Service (1997). "Eliminating the Language Barrier on the Web. A New Language Translation Service for Web Content Now Available on Digital's AltaVista Search Service", White Paper.
- Bennett, Winfield Scott (1996) "Learning from Users", in *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal, Quebec, pages 229–231.
- Choi, Sung-Kwon, Taewan Kim, Sang-Hwa Yuh, Han-Min Jung, Chul-Min Sim and Sang-Kyu Park (1999) "English-to-Korean Web Translator: "FromTo/Web-EK""", in *Machine Translation Summit VII '99: MT in the Great Translation Era*, Singapore, pages 432–437.
- Flanagan, Mary (1996) "Two Years Online: Experiences, Challenges and Trends", in *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal, Quebec, pages 192–197.
- Flournoy, Raymond S. and Christopher Callison-Burch (2000) "Reconciling User Expectations and Translation Technology to Create a Useful Real-World Application", in

- Translating and the Computer* 22, *Proceedings of the Twenty-second International Conference on Translating and the Computer*, London, pages not numbered.
- Gerber, Laurie (1997) “R&D for Commercial MT”, in *MT Summit VI: Machine Translation Past Present Future*, San Diego, CA, pages 94–97.
- Gerber, Laurie and Jin Yang (1997) “Systran MT Dictionary Development”, in *MT Summit VI: Machine Translation Past Present Future*, San Diego, CA, pages 211–218.
- Lange, Elke D. and Jin Yang (1999) “Automatic Domain Recognition for Machine Translation”, in *Machine Translation Summit VII ’99: MT in the Great Translation Era*, Singapore, pages 641–645.
- Miyazawa, Shinichiro, Shoichi Yokoyama, Masaki Matsudaira, Akira Kumano, Shuji Kodama, Hideki Kashioka, Yoshiko Shirokizawa and Yasuo Nakajima (1999) “Study on Evaluation of WWW MT Systems”, in *Machine Translation Summit VII ’99: MT in the Great Translation Era*, Singapore, pages 290–298.
- Nakayama, Keisuke and Akira Kumano (1999) “Collection of Dictionary Data through Internet Translation Service”, in *Machine Translation Summit VII ’99: MT in the Great Translation Era*, Singapore, pages 586–592.
- Story, Henry (1998) AltaVista internal presentation.
- Westfall, Edith (1996) “Legal Implications of MT On-line”, in *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal, Quebec, pages 231–232.
- Yang, Jin and Elke Lange (1998) “Systran on AltaVista: A User Study on Real-time Machine Translation on the Internet” in David Farwell, Laurie Gerber and Eduard Hovy (eds) *Machine Translation and the Information Soup*, Berlin, Springer, pages 275–285.

CHAPTER 13

How to evaluate machine translation

John S. White

PRCNorthrop Grumman Information Technology, MacLlean, VA

1. Introduction

Evaluation has always been central to the consciousness of those involved in the field of Machine Translation (MT). Historically, evaluation has proven difficult, traumatic, at times misleading, but very often both revelatory and helpful. Originally, it was the apparent results of evaluations themselves which made the general public aware of the potential for MT. Today, there is emerging a legacy of actual production use of the output of MT from which the fuller understanding of its potential becomes apparent to actual users. The increased usage of MT, however, demands more comparability and relevance among the many attributes and measures of MT.

In this chapter we will explore evaluation, to come away with an idea of why it is so central to MT, why it is so difficult to do, and why there must be many different types of evaluation for many types of users and uses.

2. The role of evaluation in MT

Three reasons for the primacy of evaluation in the MT field come to mind. We will call the first one the “Macy’s Thanksgiving Parade syndrome”: in the USA we broadcast a local parade on national television because it was once a miraculous thing to do, and has just become a tradition. By this analogy, MT evaluation is central today because it was central once, and we just focus on it today because we always have.

2.1 Tradition/trauma

Of course we will dismiss this theory, in light of the compelling nature of the other two reasons why MT evaluation is so pre-eminent. But let us dwell for just a

moment on the historical aspect of MT evaluation. MT is one of the very first applications attempted on a digital computer, having been contemplated in the 1930s and 1940s, and then implemented in large scale in a 1954 experiment.¹ In those times the demonstration of the feasibility of doing any human activity by a computer was exceptionally newsworthy, and spoke not just of the potential for a particular application but for the overall potential of automatic digital processing. The demonstrations of early MT were inherently assessments of feasibility, and at the same time appeared to be portents of the amazing promise of computers.

It is probably not the case that we focus on evaluation because of this early visionary tradition. But the pre-eminence of evaluation was driven home in a more traumatic way in 1966. The report of the Automatic Language Processing Advisory Committee (ALPAC 1966)² was used to shift the focus of government-sponsored research away from MT and into artificial intelligence and natural language processing.

ALPAC made nine recommendations for future US government-sponsored research in translation. These advocated such things as speeding up the overall translation process, development of translator aids, and so on. Three of the nine, however, directly recommended further work in evaluation: new evaluation methods, evaluation of quality and cost of translation, and evaluation of the speed and cost of machine-aided translation. These areas resonate today: as we will see, coming up with reliable, efficient, and reusable methods is difficult; and parameters of speed, quality, and cost of the MT process drive any modern consideration about whether and how to adopt automated components in the translation process.

The computing world was of course a very different place when this report was written. In particular, there was no efficient way to use flawed MT output when it only existed as some sort of printout, all uppercase. The independent development of word-processing applications has made imperfect MT more useful than was imagined then. Nevertheless, the ALPAC report continues to resonate, not only for its impact on history, but also because it was a well-written document whose observations and methods continue to have value.

So while we cannot say that MT evaluation is important today just because it used to be important, it is nevertheless of great value to regard the issues raised in the earliest days of the field as remaining critical today.

2.2 Importance

The second reason why MT evaluation is pre-eminent is because it remains important today. It costs a great deal of money to research, design, and implement an MT system, and more time and money still to complete the system with “knowledge” (e.g., words, phrases, meanings, contexts, etc.) germane to the subject areas which

the system will translate. The different interests in MT (which we discuss below) need to know whether the investment is worth making for their individual objectives.

The motivation for the remainder of this chapter is a combination of the awareness of the second reason and a third reason, which is simply that MT evaluation is hard to do. Translation is special among the set of automated applications that we may call the “human language technologies” (ARPA, 1993; ELRA, 1998), because “correct translation” is an elusive target, and because there are a range of people, purposes, and types of MT that each need different measures to indicate what each needs to know about MT systems.

2.3 Difficulty

It is axiomatic that evaluation measures some attribute of something against a standard for that attribute. For this to happen, there needs to be an identifiable “correct” or “best” ideal, whether explicit or implicit, against which to compare the relevant attribute of the individual item being measured. The most obvious standard for MT, i.e., the “right” translation, is the very thing translation itself cannot provide.

Translators, more than anyone else, are aware of the fact that no document is ever translated the same way by two different people. Moreover, if we were to take many human translations of the same document, we would find in addition to the fact that they are all different, that some seem rather poor, some seem better, and some might seem very good. What we will not see is the case that exactly one translation is exactly right. That is, given a sufficient number of translations done by competent translators, there will be a set of them about which there will be disagreement as to which is the best. And it is just as likely that not one of them will achieve universal acclamation as perfect.

So there are many ways to translate the same thing, and reasonable translators will disagree about which way is best. These facts are a testimony to the rich variability of language and remarkable creativity that goes into the act of translating. But it certainly makes life harder when trying to evaluate MT systems.

Now let us imagine that we want to evaluate some other human language processing system, say, speech recognition. Typically, speech recognition takes input spoken into a microphone connected to a computer, and produces a text (or some other representation) of what was said (this is of course a simplification over a wide range of types of speech-processing systems). There is in fact a considerable body of work on evaluation methods in speech recognition (see Hirschman, 1998). The developers need to know what parts of the answer were right. They need to know whether and why the system picked the way it did from a variety of hypotheses that it generates, what it left out, what it put in that was not there, how long it took, and

so on. However, the ultimate evaluation metric is straightforward: Did the system display on the screen the words I said into the microphone? It is possible for it to have gotten it exactly right, and everyone knows it did from having heard what was said and seen what was displayed.

MT evaluation is harder than this. Only people who know both languages can know just by looking whether it got a translation right. And as we noted above, there is great latitude for disagreement about what constitutes “exactly right” in translation. So we cannot take full advantage of the notion of “ground truth”: the set of right answers that form a universally agreed-upon standard for comparison of evaluation results (e.g., the answer key of a school quiz, or the map of a minefield). Therefore we must somehow accommodate some highly subjective judgments about which translation might be better than which other translation.

If we cannot get around subjectivity, could we perhaps take advantage of it? After all, despite the disagreement we are likely to have about translation correctness, we still strongly agree about linguistic intuitions in everyday life. We can talk to each other, read works that are hundreds of years old, order food, and so on, with very high confidence that, despite likely differences in our cultural or cognitive models of reality, we fully understand and agree about the meaning of the expressions and the event as a whole. Could we not capture these linguistic intuitions as means of measuring MT?

The answer is yes, but not exactly in the simplest way. Let us examine three evaluation methods that attempt to take advantage of my linguistic intuition, to show that exploiting these judgments is not at all straightforward.

3. Three imaginary methods

Native speakers of English can tell immediately, without any thought, analysis, or special linguistic awareness, whether something is a felicitous English. Let us imagine that we have the responsibility for determining whether a particular “into-English” MT system actually can translate. Our assumptions going in are the intuitive ones about being able to determine felicity, and the idea that we should be able to make a general claim about the ability of a system to translate the infinite expressions of language, based on a finite test set.

3.1 Case 1: output only

Let us look at the easiest possible case (Case 1). We will simply look at some output and indicate whether it is good English. We glance at a few of the output expres-

sions, and realize that we are not going to find much that is simply good English, so we devise a way to measure “how good” an expression is. We will call this a **metric**. In this case, we come up with a scoring metric as in Figure 1.

Look at each sentence, one at a time;

EITHER:

- the sentence is completely good English;
- OR:
- the sentence is degraded by up to n errors.

OTHERWISE the sentence is wrong

Figure 1. Case 1: counting errors.

We then use this metric to score all of the output sentences, and can then express either a quantitative measure (by sentence, document, or whole test set), a qualitative measure (by characterizing the errors in some way), or both.

What is good about this is that it does take advantage of our linguistic intuition (which for all its subjectivity, has a high degree of agreement among speakers, as we noted above). We do not need any particular skill to do this, except perhaps to come up with an apparently consistent, if informal, characterization of errors.

The main thing wrong with this method, though, is that we do not know anything about where these expressions came from. Are they really translations of anything? Generally, that is not the real concern, but something like it is: Can we really characterize the errors unless we know what the input is? We can tell what is wrong with the target-language output, but that characterization may not help me improve the system.

Consider example (1) from an English–Spanish system from the early 1980s.³

- (1) **no hace el conductor mas ambos*

Looking only at (1) in isolation, we do not know what to make of this. It does not make enough sense for us to even count errors, using the pure algorithm of Figure 1. So let us fudge a bit, and see what output we should have expected (2). This is a common sign found in Latin-American buses.

- (2) *no molestar al conductor*
 not disturb to-the driver
 ‘Do not disturb the driver’

Here, there are at least six errors:

- the wrong verb (*hacer* vs. *molestar*)
- the wrong mood of the verb (indicative vs. imperative or infinitive)

- the wrong morphological form of the article *el* (vs. *al*)
- the wrong grammatical assignment (*conductor* as subject vs. object)
- *mas* is an unanticipated word ...
- ...as is *ambos*

If we are examining only the target language (Spanish), we would have to conclude that there may be a lot of things wrong with this MT system. But this is not really the case. The English input for this was (3).

(3) Don't bother the driver.

The sole cause of error was that *bother* was not in the lexicon; the system tried to find a translation for *bother* by constructing it as a comparative adjective (*both* plus *-er*), hence *mas ambos* 'more both'. So in reality, this system only had one thing wrong with it (easily correctable, by the way), where our proposed metric showed six things wrong.

We will see below that monolingual judgments of output actually are very useful indeed, but we will be looking for something different, and have some better ways to capture the judgments. In the meantime, we have learned at least that we cannot yet make the general claim about how well this system translates with just our judgments and the metric in Figure 1.

3.2 Case 2: input and output

So in Case 2, we look at both the input expressions and the output expressions. The objective is still to make the infinite claim based on the behavior of a finite set of examples.

The first thing we notice is that we need to change our metric. This is because, now that we see both sides of the translation, we realize that there are two parameters, or **attributes**, that we must consider: whether the output is fluent English, and whether the information in the source is conveyed in the target English. The first measure is basically the same as in Case 1, except that now we have a better idea about where the expressions came from. The attribute of target-language fluency is known as **intelligibility**. The second parameter, the information conveyed, is known as **fidelity**. So now our metric is as in Figure 2. Now we can express the measurements quantitatively and qualitatively, along parameters of intelligibility and fidelity. Fidelity and intelligibility are, of course, correlated: a completely unintelligible expression conveys no information. This configuration helps us to solve the major concern that arose in Case 1, namely the issue of being able to tell something about the translation issues from looking at both the source and target language.

There are some negatives here as well, however. The most obvious one is that

Look at both the input and the output of each sentence;

EITHER:

- the sentence is a completely good translation
- it seems to be good English
- it seems to say just what the source language said;

OR:

- the sentence is degraded by up to n errors (intelligibility);

AND/OR:

- the sentence is degraded by up to m information errors (fidelity).

OTHERWISE the sentence is wrong

Figure 2. Case 2: intelligibility and fidelity.

we have moved away from our original going-in position that we should be able to judge a translation just by looking. Now we have to be a special sort of person, specifically a translator, to apply these metrics. This limits the portability and reusability of the measurement, by requiring special skills that may be hard to find and commit for the task of applying these metrics.

Another negative is that we still do not have a good way to make a general claim about the MT system, that it can now cover indefinitely many constructions that it might possibly encounter in actual operation. It may “accidentally” get right the sentences we have in our sample, which tells us nothing about how “extensible” the system is to the general (infinite) case. In our discussion of internal evaluations, below, we will talk about the difference between “glass-box” and “black-box” evaluations, with a view toward teasing out some of the issues of extensibility.

3.3 Case 3: input, two outputs

Of immediate interest to us, meanwhile, is whether the system can improve its coverage, whether to extend to new cases or finally get right the sentences it currently does not. So now let us examine a third case of our (suddenly complicated) method of being able to tell just by looking.

Here, we will look at the input and two outputs: one from before a particular improvement was made, and one after. The usefulness of this is obvious if we want to verify that a change we made in the system made the intended things better, and nothing worse. So now our metric is something like the one in Figure 3.

The advantage of this approach is that we know something more about our system, in particular, that it is better (or worse) than it once was. At the same time we have some notion about the extensibility of the system, since by making some changes the system now covers more (or fewer) naturally occurring linguistic phenomena.

Look at the input sentence, along with the “before” output sentence and the “after” output sentence;

EITHER:

- both translations are perfect in fidelity;

AND/OR:

- perfect (and possibly different!) in intelligibility;

AND/OR:

- one differs from the other by n fidelity errors and/or m intelligibility errors;

OR:

- one is wrong and one is not.

OTHERWISE both translations are wrong

Figure 3. Case 3: before and after.

On the negative side, though, is the fact that we may not care about incremental improvement: if we are buying a system, we want it to work now, not after it has been fixed. But this is not the main negative that has arisen now. The real problem has been there all along, but it has become critical now that we are looking at more than one version of a translation, and looking at at least three times as much data as we were looking at in the first case. Now we must confront the “human factors” biases.

As we noted earlier, we are taking advantage of the fact that our linguistic intuitions allow for a great deal of agreement among different times, places, and between different speakers of the same language. But there are local, almost microscopic effects that can lead us to inconsistent judgments and inaccurate conclusions about the results. Here are three of the classic ones⁴ that affect our current case:

History. Things outside the world of the judgments we are making can intervene. For example, let us say you and I split the task of comparing the two outputs. I do my judgments mid-morning the day after the World Cup final, and you do yours in the late evening after a stock-market crash. We can be sure that these events have influenced us, but we do not know whether they have influenced our linguistic judgments, or influenced mine more than yours (consider, for example, that a hurricane is approaching Florida on evaluation day, I have a relative there, and you don’t).

Testing. Evaluators have a different reaction to something the second time they see it than they had the first time. This prevents them from making easy comparisons between two translations of the same expression: the second time they see it, they have an informed idea of what the expression is supposed to say, and this affects their judgment of whether it actually says it or not. Moreover, experiencing a really badly translated expression will make the next expression they judge seem better than it really is, and vice versa.

Another “testing” effect is that judges will react differently to a translated expression if they (think they) know how it got that way. In human translation, one’s opinion of a particular person may bias one’s assessment of their translations; in MT, judges will be more forgiving of particular errors if they think their cause is a trivial bug (e.g., missing lexical item) rather than a serious problem (e.g., scope of modification). Now sometimes this bias is benign, in exactly those situations where someone really does understand a system architecture well enough to know the cause of an error, and is looking at exactly that phenomenon with the aim of fixing it, and knows how to generalize beyond that instance to the appropriate fix. This practice is common in early development of systems, but it should be clear that its results cannot be generalized as a claim about the quality of the translation system compared to others, or in the context of an actual intended end-use. And even if we are willing to squint our eyes at this specific bias, we must not forget that the practice of a single programmer judging output phenomena is also subject to all of the other biases described here, diminishing further the value of the claim that the programmer will make.

Maturation. Not only do things happen during the course of an evaluation, but very ordinary things can affect someone’s ability to be consistent in their judgments. Specifically, they will get tired, bored, hungry, or fed up with the process of evaluating, and so the sentences they graded later in the cycle will get a different look than the ones they graded earlier.

3.4 The verdict on intuitive judgments

We have now exhaustively considered the easiest case in MT evaluation, i.e., the prospect of simply using our intuition to make evaluative judgments. We have seen that we must see more than just the output, because we have to know about both the *fidelity* and the *intelligibility* of the translation. We have to be able to tell whether the results of a finite set of test sentences allow us to make a claim about the coverage of the range of linguistic phenomena that the system will encounter. And, since the measures we apply are ultimately subjective, we are constrained by a range of human frailties.

4. Evaluation for MT stakeholders

How then can we evaluate at all? The prospect is not at all hopeless. The answer lies in controlling the factors that we can control, and optimizing the control of those that we cannot capture completely. The largest best control we can impose is to

make the common-sense distinction among the different things that different people need to know about MT systems; in other words, that no one evaluation method will fit all needs.

The obvious, and probably most important people in the world of MT, as it is practiced today, are translators, and the people who need the translations — the information consumers, if you will. There are several other groups of people, however, who have a stake in the success of one or more aspects of MT. It is convenient to divide these into end-users, managers, developers, vendors, and investors.

4.1 End-users

- *Translators:* Translators need MT systems which are easy to access and use, compatible with their computerized environment and work processes. The system must enable translators to make the best possible use of their expertise and experience, to increase the quantity of translations they can do, and ideally also enhance the quality of both translations and the work-a-day lives of the translator.
- *Translation editors:* In the professional translation environments where post-translation editing is part of the work process, the editors have the specialized requirement of making sure that translations are both accurate and consistent with other translators' work in the same document set. Current MT systems impose additional editorial requirements on these people (and on every translator); these requirements should be easy to meet, and the system as well should make the pre-existing job of quality control, version control, and consistency easier as well.
- *Monolingual information consumers:* These are all of us, who need information at one time or another that is comprehensive, relevant, and timely, with little or no regard for the language of its origin. Here, the work of the MT system in the overall flow of automatic information processing should be transparent to the information consumer.
- *Office automation users:* We are more and more accustomed to inter-operating suites of applications. The days of single purpose, "turnkey" computers are long gone. If we operate MT systems in the course of our work, we should expect them to accept input from other office automation (OA) applications, and return output compatible with those other applications in the OA suite.

4.2 Managers

- *Operational managers:* These need to know if an MT system will work in the environment of their translator employees, given the environment, requirements,

etc. The operational manager needs to know whether MT will actually improve the performance of the translation department. The JEIDA evaluation methodology that we will present in Section 5.5.1 is designed to help people make this sort of decision.

– *Procurement managers:* The people responsible for purchasing systems need to know whether the system requires equipment or connectivity that the department presently does not have, or requires special licenses or usage costs that might have to be taken into account. They also would like to know whether the company that provides a system is sufficiently sound and viable to provide support and upgrades.

4.3 Developers

– *Researchers:* There are many types of research, of course, with different objectives and at different levels of maturity. However, a common need in research is to know whether a particular approach actually matches the hypothesis for its success. Other issues in research have to do with the extensibility of a translation approach beyond a particular set of phenomena into the infinite world of real language use.

– *Productizers:* The people who take the fruits of research and attempt to make a marketable product need to know whether the conceptual prototype can ramp up to meet the needs of real use, and can fit into a real automation environment.

4.4 Vendors

Vendors need to know whether an MT system they wish to sell is robust and extensible enough to fit into a variety of different settings, i.e., if the demand is sufficient to justify the marketing and support investment.

4.5 Investors

– *Research organizations:* Organizations such as government agencies that sponsor research need to know whether sufficient progress is being made to demonstrate the research hypothesis, and that results that do appear are not artefacts of extraneous effects. The DARPA methodology we will discuss in Section 5.6.1 is an example of an evaluation method for this purpose.

– *Venture capitalists:* People interested in investing in high technology in general need to know whether MT is a worthwhile endeavor. Here, they need to know whether the technology is in fact viable, whether the companies trying to develop and market it are stable, and what the future trends for demands and state of the art will be.

5. Types of MT evaluation

As we noted, the different responsibilities and obligations of each of the stakeholder groups means that each group needs to know different (though often overlapping) things about MT. The end-user in a translation environment does not need to know what the cost of a system is (unless that person is also responsible for procurement or management) in order to do their job. Nor does the end-user have to know where in the analytical engine pronominal reference is handled, unless the user interface is rather primitive. The investigator of a particular scientific approach to MT is unlikely to be concerned, at first, about whether a system that will someday incorporate the results of their findings will run efficiently on a conventional desktop computer. Thus each stakeholder's need for information must be covered by a particular, pertinent, set of evaluation types.

Here we will lay out a descriptive model of evaluation types, and then devote the remainder of this chapter to a discussion of each type along with some of the fundamental issues that arise for each type of MT evaluation, and some examples of evaluation methods. A convenient organization of types might be the following:⁵

- Feasibility tests
- Internal evaluation
- Declarative evaluation
- Usability evaluation
- Operational evaluation
- Comparison evaluation

We will now expand on these evaluation types and illustrate the various ways the issues we have discovered are addressed in some classic approaches.

5.1 Feasibility evaluation

The very first glimpse the general public got of MT was essentially the result of a feasibility study, that is, an evaluation of the possibility for a particular feat to be accomplished at all, or for a particular approach, whether it has any actual potential for success after further research and implementation. Feasibility evaluations provide measures of interest to researchers and the sponsors of research.

Let us say that we have studied a particular linguistic theory, or perhaps a particular method in computer science, and it occurs to us that we might be able to apply this new finding to the automatic translation of languages. Our supposition is probably naïve, but nevertheless we need somehow to demonstrate that it is at least possible to think about translation using this new approach. We must show something translated by this approach, but beyond this we must be able to make a

prediction that this approach is actually of use in meeting a particular translation objective.

At this point we do not need to show that we can handle every possible construction in the source and target languages. We do need to show two things, however:

- that we can handle certain well-known contrasts between the source and the target in a way that promises extensibility to more complex instances of the same phenomena; and
- that we can do this because of (rather than in spite of) our new approach.

So we can refer to the **attributes** of feasibility testing to be coverage of representative sub-problems, and extensibility to more general cases.

Of all the evaluation types that measure something about the quality of the translation output, in feasibility testing we can come closest to using something very much like the “ground truth” we earlier said was impossible. Since we are only trying to show that it can do a very bounded set of sub-problems very well, we can strictly craft our test set to control everything in each test expression except the phenomenon being tested. So in this sense we will actually be looking for a single “right” (ideal) translation for each such expression.

Where do we get the test sets? We should develop simple source-language patterns that are “theory neutral”. This means that we will be looking for contrasts between the source and target language that are the sort of thing that anyone would agree are different between the two languages. So, for example, the typical order of nouns and adjectives is different in English and French, regardless of what theory we use to explain the phenomenon. It is this sort of phenomenon that is going to show us (and our funding sponsors) in a visible, understandable way that our new approach has potential. Probably the best place to get descriptions of contrastive phenomena is from a pedagogical resource, for instance, a contrastive grammar of the two languages written for language teachers.

There is another set of issues we must cover with the feasibility test. We are essentially claiming that a particular linguistic and/or computational and/or implementation approach will do translation somehow “better” than existing approaches. So in addition to showing fundamental coverage of the contrastive issues, we must also show two other things about our approach: that its good points facilitate the coverage results, and its bad points (particularly the troublesome linguistic theories) do relatively little harm.

Suppose, for example, that our underlying linguistic theory requires that semantic representations are expressions of some logical form that transcends individual languages. This theory seems to presuppose a “metalanguage” for these expressions, which should facilitate interlingua-type MT implementations, except that our

theory also insists that the logical form uses the same metalanguage as the structural descriptions of the source-language analysis and target-language synthesis. In the feasibility test we must show that the interlingua implementation model is indeed made more effective by the application of this theory, and that adherence to representational purity does not make generation of target structures too difficult.

5.2 Internal evaluation

Internal evaluation occurs on a continual or periodic basis in the course of research and or development. Here, the question is whether the components of an experimental, prototype, or pre-release system work as they are intended. The particular items covered in such an evaluation will vary with the maturity of the system being evaluated of course, and thus provide measures of interest to researchers, research sponsors, developers, and vendors.

As with the feasibility test, we want to be able to show that we can cover the fundamental contrastive phenomena of the language pair. But we need to show some other attributes as well, namely that the system we are developing, or bringing to market, or adapting to our own user environment, is improving. We need to show, for instance, that as we add grammar rules, or dictionary entries, the system translates the things we are trying to improve better than it did, and does not suddenly fail to do something it used to do. So we need to have a standard set of test materials for **iterative testing** (tests designed to make sure an improvement in one area actually works and does not adversely affect another area).

We need to show some other things at this point. In feasibility testing, we were concerned about showing the potential of an approach, and therefore needed to demonstrate certain very focused patterns corresponding to the obvious contrasts between two languages. In internal evaluations we must show that the implementation of our approach can also extend beyond these patterns into the language text that will actually occur in production. So internal evaluation typically handles both: the patterns for regression testing against specific phenomena, actual text for determining extensibility.

The imaginary methods we discussed earlier in this chapter — trying to use our common sense and intuitive command of English to evaluate MT — were cast in the metaphor of an internal evaluation, though the issues we discovered there are also germane to the declarative type of evaluation we discuss below. But we have already alluded to one issue that is very relevant to internal evaluation, namely, how we tell from our test sample whether we can really claim to cover the literally infinite variation in source-language phenomena. Fundamentally, we have some input, some output, and (in the present case of internal evaluation) a design in which the system's components do distinct things that come together in the intended way to

produce the intended output. How we regard these elements in evaluation gives us different, and equally useful, views of internal performance and predictable extensibility.

5.2.1 Black-box vs. glass-box evaluation

The way you look at the relationship of the input and output has been referred to as the difference between “black-box” testing and “glass-box” testing. The black-box view is a look at the input and output without taking into account the mechanics of the translation engine. The glass-box view looks inside the translation engine to see if its components each did what was expected of them in the course of the translation process.

There are advantages to each. The black-box view is portable (i.e., the method and measures are external to the design and philosophy of any one system). It is more amenable to comparisons of systems, and to determining the current language coverage of a particular system. The glass-box view helps to determine the extensibility of coverage of the system, by being able to tell whether and how well the designed processes perform their functions. Did, for example, the transfer rules correctly move a prepositional phrase to the right position, or did an apparently correct result come from a fortuitous default?

Let us look at some examples of each view. In the first instance, we are looking to see if an English–Spanish system covers the common contrastive phenomenon of existential quantifiers using the auxiliary verb *haber*.

Consider first (4a) and the output (4b) obtained from the system.

- (4) a. *There is* a gun in my bedroom.
- b. *Hay un revólver en mi alcoba.*

This looks all right; *there is* is appropriately translated by the Spanish existential copula *haber* (inflected as *hay*). However, in (5),

- (5) a. *Is there* a gun in my bedroom?
- b. **Es allí un revólver en mi alcoba?*

it fails, hinting that this system only gets *haber* when the input is exactly *there is* or *there are*. This suspicion is confirmed by (6), where we do not want *there are* to be translated this way, because the two words belong to different constructions.

- (6) a. Some of the people over *there are* Spanish.
- b. **Alguna de la gente sobre hay Español.*

Using just a black-box view, we are able to measure the coverage of this system, and even have a hypothesis about how the system tries to handle these phenomena. And this set of test sentences is entirely reusable for other systems.

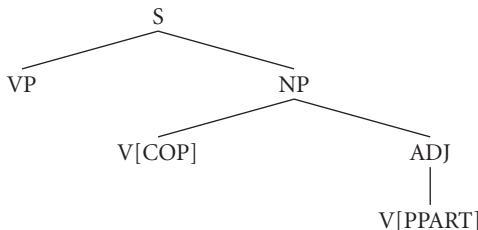


Figure 4. Internal representation of (wrong) syntactic analysis of (7a).⁶

Let us see a glass-box evaluation, this time of a Spanish–English system, which provides a trace of the processing of components of the input (7a), in this case a syntactic tree of the output (7b).

- (7) a. *La puerta fue cerrada.*
 b. The door was closed.

Here the output is a correct translation of the Spanish passive construction. But we need to look inside the box to see how it was arrived at (Figure 4).

We can see from the syntactic tree created by this system that what appears to have been correct was after all incorrect: the syntax shows that it created a predicate–adjective construction rather than a passive. This is understandable, since *fue* ‘was’ can be a copula and *cerrada* ‘closed’ can be used as an adjective just as in English, though not in this case. By being able to look inside the box, we are able to determine that despite apparently translating (7a) correctly, we know that its coverage is not reliably extensible to new instances of related constructions. For example, we can predict that the system will likely translate (8a) incorrectly as (8b) instead of (8c).

- (8) a. *La puerta fue cerrada por John.*
 b. *The door was opened (in exchange) *for* John.
 c. The door was opened *by* John.

Note that we could have come to this conclusion with a black-box view as well, but only if we had thought to test examples of passives with and without the *por* construction. So the glass-box view is better for predicting how the system might handle novel inputs, but it does not allow for direct comparisons with other systems (whose internal constructions could be radically different from the syntactically oriented one in the example, and thus not meaningfully comparable).

Much of the methodology development in internal evaluations has to do with how to characterize the errors found. In the mythical examples earlier on in this chapter we simply counted errors, but in internal evaluation there is some idea of

criticality of errors (against internal design principles), as well as mitigating the issues illustrated by example (1)–(2) above.⁷

5.3 Declarative evaluation

This evaluation is the heart of the matter for the casual observer. It addresses the question of whether a system translates well, by which is meant, among other things, the degree to which it has the attributes of fidelity and intelligibility that we introduced above. This evaluation type is clearly of particular value to investors, end-users, vendors, and managers, but also to developers.

The purpose of declarative evaluation is to measure the ability of an MT system to handle text representative of actual end use. In certain ways, we might expect the methods used here to be very much like those of internal evaluation — coverage of linguistic phenomena and handling of samples of real text, to name two obvious methods. However, here we must show more than a largely constrained test set. We are more interested in what the system can currently manage than what its extensibility potential is; therefore we may well be more likely to look at black-box views.

We have already talked about the attributes of intelligibility and fidelity, and these are the principal attributes that declarative evaluation measures. However, the degree of these attributes that are or are not acceptable depends on more than a monolithic standard. Rather poor MT can be useful for certain types of tasks. For instance, we may need to know just enough to throw something away, or to get some names and places from an article, or write a quick “gist” of its content, or determine that we need a really good translation of something. “Task-based” measurement of this sort is perhaps a necessary evil of the less-than-perfect MT of today: if all MT output were “perfect” (for instance, as fluent and informative as if it had been originally written in the target language), we could do every task with it. But for now, it proves to be a very useful delineation of the way that we interpret intelligibility and fidelity results.

Three principal methods are used in declarative evaluation:

- Analysis of errors. In declarative evaluation, these must be independent of consideration of the approaches to translation that guide feasibility testing. And this of course is hard to do, as we have seen: the problems with significance of an error count, as well as the human factors problems, have already confronted us (Flanagan, 1994).
- Rating of ability to do a task as a result of the output.⁸ These methods generally have people do some task using translated material as a guide. Care must be taken in these methods to distinguish between the test subject’s baseline ability to perform the task, the actual usefulness of the original document, and the effects of the translation process.

- Rating of intuitive judgment of the “goodness” of the translation (Nagao et al., 1985). This set of methods addresses head-on the issue of the inherent subjectivity of judgments about translation. In fact, these methods exploit the subjectivity, by dividing judgments into small segments and using a large sample of both judgments and subjects. We will summarize a classic study of this sort here (and will present a similar approach as an example of comparison evaluation below).

Additionally, important progress has been made in the establishment of standards for MT software engineering, and the evaluation metrics that accompany them (see EAGLES, 1996).

5.3.1 *Declarative evaluation: ALPAC (1966)*

This evaluation designed by John B. Carroll comes from the early days of MT, and is described in the ALPAC report we have already introduced. Carroll sought a standard method of evaluating both human and machine translation, that was simple and portable, yet highly reliable. He realized that subjective judgments about translations show promise of meeting these goals. He also realized all of the human factors that come with subjective judgments. The method he arrived at is an ingenious optimisation of simplicity and portability, while incorporating as many controls against human biases as were possible and practical.

As we know, in a fidelity evaluation, we attempt to determine how well the content of the source text was conveyed in the target text. But what if the original is not very informative in the first place? If it does not tell you much, then there may not be as much content to measure (at least of a descriptive, narrative sort). Carroll was sensitive to this possibility, and designed his fidelity method accordingly: human raters were given the task of judging the “informativeness” of the *source* document relative to a previously read *translated* version. This appears backwards, (why are we scoring the source document when we want to know the fidelity of the target?) but it allowed Carroll to factor out the inherent informativeness of the original document as a source of unwanted variance.

As we saw early in this chapter, measuring fidelity requires an awareness of the source document, which usually means a translator is required to make the judgments. Carroll sought to optimise simplicity and portability by keeping judgments monolingual if possible. So he developed two versions of the fidelity measure, one involving raters with expertise of the source language, Russian (directly comparing the informativeness of the English and the original) and one where the raters were monolingual (comparing English outputs with expert translations)

Carroll’s method proceeded as follows: Four passages from a Russian scientific document were selected and translated by professional humans and a variety of MT

systems, a total of 6 translations each (3 human, 3 MT). From these, sentences were extracted for the actual measures, and organized for presentation to raters in a way that prevented anyone seeing more than one translation of a particular sentence. Each rater saw 144 sentences.

For the intelligibility measure, 18 students who did not know Russian judged each of the extracted sentences on a nine-point scale, where each point was verified as equidistant cognitively. For example, point 1 (lowest) said: “hopelessly unintelligible...”, point 5 said “the general idea is intelligible only after considerable study...” and point 9 said “perfectly clear and intelligible...”.

The fidelity measure had the monolingual and bilingual variants, as mentioned above. Raters saw each extracted sentence, then rated the *reference* version of that sentence (original Russian for the bilingual variant and expert English translations for the monolingual) on a 10-point scale that indicated how informative the reference version is, having first read the translated sentence. On this scale, point 0 said “the original, if anything, contains less information than the translation...” and point 9 said “... makes all the difference in the world...” Remember that the raters are scoring the reference version, and therefore the lower the score, the higher the fidelity of the translation.

Results of this evaluation demonstrated that human translations are more faithful and more intelligible than machine translations of that era. This is no surprise, of course; the same generally holds today. Other findings were more surprising, however. For one thing, they appeared to show that the results of the monolingual fidelity exercise were consistent with those of the bilingual — meaning that such measures could be done with less expertise, and thus more simply and portably, than would be possible with the bilingual version. This implies that no information was lost between the original Russian and the expert translation versions. It could also be an effect of differential expertise among the Russian-speaking raters, or possibly caused by a great difference in quality among the samples used in the exercise. In any case, it is an interesting result for its suggestion that fidelity, like intelligibility, may be successfully measured monolingually.

The results also showed that fidelity and intelligibility are highly correlated. As we have noted earlier, there is an obvious convergence at the extreme (an utterly unintelligible output cannot convey any information); but these results may suggest that the correlation continues to carry significantly over many degrees of intelligibility. The implication of this result is that in the future we may be able to find a way to measure just one of the attributes (probably intelligibility) and infer fidelity from it. We do not know how to do this yet, because we must continue to account for the instances where a very readable translation happens to give the wrong information.

5.4 Usability evaluation

Even if all other aspects of an MT system work as advertised, it will never actually do any good unless what it can do is actually accessible to the people who will use it. Measures here have to do with common standards of response time, number of steps to complete a task, etc. Developers and vendors will find value in these studies.

The purpose of a usability evaluation is to measure the ability of a system to be useful to people whose expertise lies outside MT *per se*. As we have described the user set above, these people may be translators, editors, analysts requiring a particular type of information, or any other sort of information consumer. There are common expectations about how a computer application should function (some of which overlap with operational criteria discussed below), and these can be measured rather broadly across user groups. However, there are usability issues that are specific to the type of job a user does, and indeed to the specific environment in which the user does it.

The usability of a system is a function of two attributes, the **utility** of an application and the users' **satisfaction** with it (see White and O'Connell, 1996). There are quantitative metrics that can be associated with either of these factors, but often there is great reliance on users' subjective assessments. Naturally, usability is measured at the point of interaction between the user and the thing being used, in this case the MT software application, and this means that the focus of such evaluation is on the apparent functioning of the user interface. Evaluation of interface properties may include:

- the time to complete a particular task
- the number of steps to complete it
- how natural the navigation process appears to be
- how easy it is to learn how to use the application
- how helpful the documentation is

The quantitative measures for these may include timing the particular processes, counting the number of steps a user goes through to complete a task (apart from the number of steps the developer intended), and the counting errors users make during operation. Note that all of these measures must be controlled very carefully for the human factors biases we have already discussed: For example, we have to make sure we can attribute user errors to the usability of the system and not to some characteristic of the individual user. We do this by getting as large a sample of users as we can, by controlling the sequence of things we measure, and by making the circumstances for each user as alike or as analogous as possible.

Subjective measures may include questionnaires about each usability aspect of the system: e.g., adaptation, operation at run-time, maintaining the data files, online

help, and so on.⁹ Of particular value is narrative feedback from users about what they think of each step in a process while they are doing it. The user narratives may provide valuable information about needed system improvements.

5.5 Operational evaluation

Operational evaluations answer the question “Is it worth it?”. Here, the primary factors to consider are all of the costs involved, against all of the benefits. Issues like common platforms and operating systems are germane here. End users and their managers need these evaluations, and thus investors and vendors must be attentive to the operational factors.

The purpose of operational evaluation is to determine the cost-effectiveness of an MT system in the context of a particular operational environment. Some operational considerations appear similar to usability issues, and in some sense they are: If an otherwise functional and compatible MT system is not used, you will have lost money by buying and installing it. However, in general there is a different point of view. If, for example, the way to save a file in an MT application is unlike the way to save a file in all the other OA applications in use, the usability impact has to do with a loss in utility and time to perform a task. From the operational point of view, the same property affects throughput time (time to produce a certain quantity of translation), quality degradation, and perhaps training cost.

A meaningful measure in operational evaluation is **return on investment**, which implies comparison of the measurement of the real costs of an MT application, and the real benefit (revenue, cost savings, etc.). We then may compare the value of these properties against the same measurements of the way the process is currently done. The result is an expression of the benefits of inserting MT technology (or not), expressed in terms of the attributes of productivity, cost, revenue, or quality.

Among the factors to be considered in measuring these attributes are these:

Operational environment

- compatibility with the familiar (Does the MT software (appear to) run on my desktop computer?);
- compatibility with the standard formats (Does the MT system accept input from, and output to, the OA formats I use everyday?);
- consistency of the application GUI with the operating system (Are the common toolbar items in the same place in this application as they are in the other applications I use?);
- response time (less an operational issue than it once was, and perhaps more of a usability issue: Does it have roughly the same response time as

the other applications I use?);

- humans in the loop (Does this application require human intervention to prepare/correct data, or to operate the application?);
- preparation, throughput, correction, and output times.

Application Design

- extensibility (Does the system have a user-accessible lexicon, or other ways to customize for this environment?);
- use of standards (e.g., Does it handle the common codes for writing systems?);
- number of steps to complete a task (i.e., the number of steps designed or recommended);
- fail-softs (Does an MT failure cause an exit from the program? Does it cause a system crash?).

Provider

- documentation (Is it complete and helpful?);
- support (Is the support timely and adequate?);
- improvement (Are there periodic new releases? Do they fix user-discovered bugs?);
- corporate situation of provider (Will the provider be around long enough to support the product system through its life cycle?).

Cost

- of the system (hardware, software, licenses);
- of maintenance;
- of the process (both the automatic parts and the human intervention parts);
- of human translation (i.e., Does the overall MT process wind up being cheaper than professional human translation?).

5.5.1 *The JEIDA Evaluation (1992)*

A recent effort to capture both the operational factors in selecting an MT system, as well as the technical factors that feed into the development and user decisions, was undertaken by the Japan Electronic Industry Development Association (JEIDA). JEIDA was confronted in the early 1990s with an explosion of industrial/commercial translation requirements, beyond the scope of human translation. Yet JEIDA was cognizant that MT quality was not yet to the level that MT could deliver suitable translation for all document types and for all needs. The evaluation methodologies developed by JEIDA focus on the awareness that we have already discussed, that the different stakeholders of MT need to see different things about MT systems. JEIDA devised a comprehensive set of questionnaire materials that covered several views each of the needs of users, production managers, research and development, and research managers and sponsors. As significant as the exhaustive

coverage of the range is MT system issues is the highly visual means they chose to represent results, which can tell stakeholders at a glance both properties they need in a system, and how those match with the strengths of particular systems.

The study developed comprehensive questionnaires for each of three criteria:

- user evaluation of economic factors: which type of MT system would net the most economic benefit — as part of the decision process of whether to introduce MT into an environment;
- technical evaluation by users: which type of system will best fit the needs of the environment — after the decision to introduce has been made; and
- technical evaluation by developers: where is a particular system now compared to its own objectives for coverage, accuracy, ease of use, etc.

For economic factors, extensive questionnaires covered areas such as current translation situation (including quantity, formats, language pair), organization (how translations are done), cost, quality, and turnaround time required. These questions were then associated with 14 parameters:

- A1: Present translation needs
- A2: Type of document
- A3: Quality of translation
- A4: Language pair
- A5: Field of application
- A6: Time
- A7: Automation
- A8: Organization
- A9: Cost
- A10: Text data extraction
- A11: Re-insertion of text data
- A12: Installation conditions
- A13: Pre-editing
- A14: Post-editing

These parameters in turn form the basis of modelling the user's environment (current situation and needs). The model is amenable to representation known as a “radar chart”, which imparts an immediate, visual representation of all of the parameters at once, as illustrated in Figure 5.¹⁰

Meanwhile, MT systems are categorized by “type” (e.g., batch in-house translation, high-quality translation, interactive human-assisted translation, terminology bank, word processors, etc.). These MT types each have properties that correspond to the same 14 parameters as those of the user's environment, and so each type is also amenable to representation as a radar chart, as for example in Figure 6.

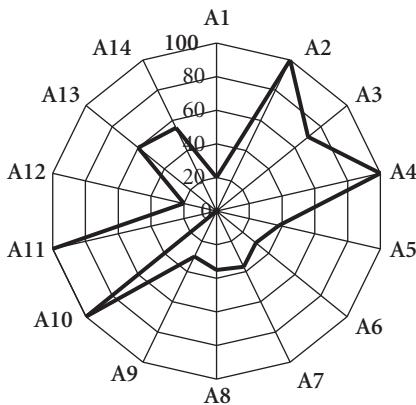


Figure 5. Example of a radar chart resulting from a questionnaire.

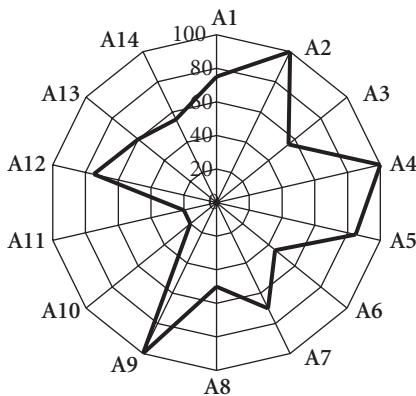


Figure 6. Example of JEIDA radar chart corresponding to a given system type.

By this means, a user's situation and a system type can be matched almost at a glance, by comparing the configuration of the radar charts. For the criterion of technical evaluation by users, factors like quality of translation, introduction costs, pre/post-editing, dictionary etc., will vary in importance depending on how the system will be used. Thus there are formulas for the composition of questionnaire scores that are sensitive to the different intended end-uses and requirements (such as timeliness of translation), along with an MT system provider's self-assessment, and preliminary evaluation of a system's output. The result is a radar chart for a system, reflecting the appropriate weightings for the user's environment. The radar chart represents both the performance of a particular system, and the particular user's satisfaction with it. It is a picture of performance whose peaks and valleys are gauges relevant to the specific needs of a user group.

The criterion of technical evaluation by developers is an in-house evaluation of the technical level the systems has achieved, and whether it has met its internal development objectives. In this criterion, fundamental properties of a system are covered in a questionnaire for researchers, decomposed into components such as dictionary, analysis and generation methods, and operating environment. These components in turn are associated with parameters such as subject domain dependency, openness to the user, and ease of operation. As with the other criteria, the resulting radar chart gives an immediate comparison of the current state of a system and its target state.

5.6 Comparison evaluation

Comparisons measure some attribute of a system against the same attributes of other systems. Thus the methods of comparison are the same as the methods of the other evaluation types, applied among several systems. This is of obvious benefit to purchasers of systems and investors in system development and productization.

The purpose of comparison evaluations is to determine the best system, best implementation, or even the best theoretical approach for meeting current or future needs. It appears that comparison evaluation can measure the same attributes as the feasibility, internal, operational — in fact, any of the other types. Depending on what we are comparing, it has all of the properties of any of these other types, except that in each case we are holding the measurements of one against the same measurements of another. For example, counting errors can be used to compare systems by errors produced, with all other factors optimally controlled (Flanagan, 1994).

Similarly, operational and usability characteristics may be compared among systems, analogous to *Consumer Reports*.¹¹

For this same reason, comparison evaluation can use any of the same methods as the other types. The caution is of course that the methods have to make sense for what it is we are comparing. So, for instance, the methods of the JEIDA study above can be readily used to determine which system of several possible candidates to select, but could not tell you which of two linguistic approaches gave the best results for prepositional-phrase attachment.

5.6.1 Comparison evaluation: the DARPA series

We will illustrate this type of evaluation with a method currently in use that employs a declarative evaluation methodology to compare systems that may be very unlike each other (White et al., 1994).

During the 1990s, the US government Defense Advanced Research Projects Agency (DARPA) developed a set of methods for evaluating MT which sought to

express meaningful measures of the performance of the system prototypes of its three funded MT projects. There was a big problem, though, namely, the three projects had very little in common. Each system translated different language pairs (French, Spanish, and Japanese into English). Each system envisioned a different end-use environment (automatic batch translation vs. human-interactive translation vs. authoring tools). Finally, each project had radically different theoretical approaches to translation, from purely statistic to purely knowledge driven, and points in between.

As a sponsor of research, DARPA needed to be able to see through all of this diversity to be able to determine whether the “core translation” capabilities of each system showed promise as a breakthrough for the next generation of MT approaches. Regardless of the approach to the translation process, every system has some component that renders a string of the source language into a string of the target language. This is the “core translation algorithm” that the DARPA methods try to measure.

The DARPA methods could not take advantage of any linguistic phenomena (because of the different pairs involved), or anything in common about the system’s approaches (since the approaches are so different). This was the ultimate “black-box” requirement. The DARPA methods used the judgments of target native speakers, who did not know the source languages, to make a variety of judgments about intelligibility and fidelity through three exercises:

- **Adequacy:** this is a fidelity measure intended to capture how much of the original content of a text is conveyed, regardless of how imperfect the English output might be. In this evaluation, expert human translations were divided up into syntactic “chunks”, and then arranged side by side with a system translation (without any chunks). The English speakers (“evaluators”) were asked to look at each fragment, and indicate on a 1–5 scale the degree to which the information in the fragment is present in the translation. Figure 7 shows an example.
- **Fluency.** This is an intelligibility measure, designed to determine how much like “good English” a translation appears to be, without knowing anything about what information is supposed to be there. Here, evaluators used another 1–5 scale to judge documents a sentence at a time. An example is shown in Figure 8.
- **Informativeness.** This is another fidelity measure, used to determine whether there is enough information in the translation to answer specific questions about its content. Evaluators answer multiple-choice questions about a translation rather like a reading comprehension test (except that we are testing the reading and not the reader).

In the largest exercise of the DARPA method, the research systems were joined by several mature commercial and institutional MT systems, and expert human translations as controls. Each system translated 100 general newspaper articles of approximately 400 words. These articles were selected in a nearly random way. That is, after a random selection of newspaper articles in each of French, Spanish, and

[Funeral Service for Michael Jordan's Father]	_____	Funeral service for the father of Michael Jordan
[Family and close friends of American basketball star Michael Jordan gathered together on Sunday]	_____	The family and the near the star of the American basket-ball Michael Jordan develop are gathered Sunday for a funeral service to the memory of his/her/its father James.
[for a memorial service for Jordan's father, James.]	_____	
[There was considerable security.]	_____	The security was important and the press had been put secluded of the church Methodist épiscopaliennes Africaine (African Methodist Episcopal Church) placed nearly Teachey (Caroline of the north), where took place the service.
[and the press had been kept away from the African Methodist Episcopal Church near Teachey, North Carolina, where the service was held.]	_____	
[Reporters had received a program of the service.]	_____	The journalists had receipt a program of the ceremony, that comprehended a message of the widow of James Jordan, Deloris, and of the his/her/their five children Michael, James Ronald, Deloris, Larry and Roslyn.
[which included a message from James Jordan's widow, Deloris.]	_____	
[and from his five children, Michael, James Ronald, Deloris, Larry, and Roslyn.]	_____	

Figure 7. Example of an adequacy evaluation page, from a 1994 evaluation.

Test Number:	Your Rating:
FLU 2X	<input type="button" value="▼"/>
Saigon and Montesque Comment on the Meeting with Buwaiz and French encouragement in conducting Elections.	Your Rating: <input type="button" value="▼"/>
The president of the French National Assembly, Philip Saigon, and the vice president of the parliamentary Council for Foreign Affairs of State-owned corporations, Do Montesque, expect that there will be an absence of tangible results for several weeks from the policy of Benyamin Netanyahu towards the peace plan and they called for not abandoning Lebanon and not leaving the US to resolve all of the issues.	Your Rating: <input type="button" value="▼"/>
The National News Agency reports in its report from Paris that Saigon was questioned by the Foreign Minister Faris Buwaiz during his most recent visit to Paris on some of the details linking current preparations in Lebanon for parliamentary elections and their promoting their conduct and the discussion of the results of the Israeli elections.	Your Rating: <input type="button" value="▼"/>
The news agency reported Saigon as saying, "Lebanon decided to resume its operation after the last Israeli war despite the spiritual and material losses from the war while depending on the Lebanese people to promote unity and to join together the divisions and this in my opinion is necessary for the goal.	Your Rating: <input type="button" value="▼"/>

Figure 8. Example of fluency evaluation page, from a recent evaluation.

Japanese within a certain range of dates, articles were culled out so that only one article covered a particular topic in the news (to avoid particular biases).

Fourteen translation systems participated in this evaluation: 5 French–English, 5 Spanish–English, and 4 Japanese–English. For each language pair there were two expert translations of each newspaper article (as we might have suspected by now, these two translations are different from each other, often in intriguing ways) (see Helmreich and Farwell, 1998). One expert translation was used to develop the informativeness questions and the adequacy “chunks”, and the other expert translation was put into the mix of documents to be evaluated, as controls.

All the translations, from all the language pairs, were collected together and arranged in evaluator sets. Each evaluator performed one or more of the measures in a way that made sure that they never saw more than one translation of any particular article across all measures, that they never saw more than one output from any system in the same measure, and that they never had the output of a certain system immediately before the output of another system more than once in a measure. All these, plus the imposition of mandatory periodic breaks, were designed to control for maturation and testing effects.

The DARPA method is cognizant of the problems of judging “correct” MT, while using subjective speaker judgments to establish the three measurements. Each of the measures divides these human judgments into multiple decisions, so that each translation has from 6 to 25 or more judgments. The value for each translation is the mean of the individual judgments. The score for a system for each measure is the mean of the values of all the texts it translated. The score each system gets for each measure can readily be compared with the scores for other systems in the same language pair, expert translations (which should theoretically be as close to perfect in the scoring as we can get), and even compared with systems in other language pairs with similar maturity. And of course, these results can be compared with the results of previous DARPA evaluations, to tell us how a particular system has improved in intelligibility and fidelity.

The results of the largest exercise have been examined from a number of perspectives. Some systems did consistently better than others on all measures, and occasionally some systems actually rivalled the performance of the expert human translation. The results appear to provide indicators about the better core translation approaches, but at the same time seem to indicate that other factors, such as overall maturity of the system — whatever approach or language pair — have a lot to do with their performance on these measures (see White, 1995). For this reason, a subsequent application of the evaluation (Taylor and White, 1998) has provided, where possible, lexical information to help level the field.

Other results seem to conform with the findings of the ALPAC study we described earlier. In particular, there seems to be a correlation between all three

measures. The informativeness and adequacy are closer (as expected, since they both measure fidelity), but both these are rather close to fluency. Again, we are tempted to find a way to measure just one of these attributes and infer the other from it.

6. Automatic MT evaluation

Throughout this discussion, it has been obvious that most of the measures we might attempt on MT are subjective in nature, whether for good or ill. This implies at the very least that it takes some time: People do not do analytical things very quickly, compared to computers. Compounding this is the array of attributes we have presented here which may be relevant for a particular need, and, of course, the awareness that allowed us to realize how hard MT evaluation is in the first place, the lack of ground truth.

It would be extremely beneficial to just about all of the stakeholders if there were some device like the equipment used to diagnose a car engine, or the little dots on batteries that let us know whether they still have charge. We have already established that there is no one measure that will tell every stakeholder everything that they need to know. So automating any one measure will not effect a panacea. But there are ways to automate measures germane to several types of MT evaluation.

In feasibility testing, for example, a well-designed test set will control for all sources of variation. So we could actually develop an automatic scoring mechanism that will compare the output with the expected result, letter by letter. The caution here is that we must avoid using such a device to make the tempting more general claim that the results say something about the ultimate potential of the MT system or approach under test.

In usability, as another example, the whole point appears to be to see how users respond to the usage characteristics of the system. So it seems that there is nothing we could automate, but even here there is — there are known standards for such things as screen color and response times. These can be measured automatically, and factored into the other data collected from user sessions.

The most intriguing and challenging recent efforts attempt to automate declarative evaluations. The crucial thing to consider is this — if we know we cannot measure everything automatically, we must be able to predict from those things that we can measure automatically the behavior of the those things that we cannot.

Recent experiments in this sort of automation have focused on one of two directions: automatically capturing some aspect of an output's *fidelity* and extrapolating *intelligibility*, or starting the other way around. One might automatically

compare and count the “named entities” (people, places, and things that are referred to by name in the text), and make a declarative evaluation claim about the translation as a result of that.¹² Or one might accumulate computational “models” of the target language, which describe a kind of Platonic space of ideal (most likely) target expressions, plot some equivalent computation of the MT output on that space, and see how close it is to the “ideal”. The former type is an automatic measure of fidelity, and the latter an automatic measurement of intelligibility.

Each of these approaches has considerable potential for speeding up the declarative evaluation process. Each, however, has an implicit presumption that it is possible to predict intelligibility from fidelity (or vice versa). This means that the two are not completely independent, and that seems to be true: absolute unintelligibility (an output that is a random set of dots) conveys absolutely no source information. But the correlation between the two attributes in the range of phenomena that occurs in translation is not established. Moreover, each approach is rather easily “spoofed”, or defeated. An automatic intelligibility method that compares output to a model of good target text can be tricked by simply outputting some set text all the time. The fidelity-first approach can be tricked by outputting a list of word-for-word translations from within the test subject domain.

Some recent approaches have the potential to connect both attributes sufficiently to mitigate, if not completely eliminate, these issues. One approach known as BLEU (BiLingual Evaluation Understudy; Papineni et al., 2001) is essentially intelligibility-based, but has some power to impose some fidelity constraints. BLEU works from the hypothesis that, even though there are many different ways to translate a text “correctly”, most of them will share certain phrases in common. If this is correct, then it should be possible to model statistically a quasi ideal of that text against which translations can be compared in relatively simple string-by-string matches. This approach requires multiple human translations of the test corpus, but has the advantage of having content-bearing ground truth, thus apparently preventing attempts to game the intelligibility measure with non-translations.

This presentation does insufficient justice to the particulars of recent approaches to automatic evaluation, and to the potential of these investigations. It is likely that some attributes of MT may be adequately measured by automatic means in the near future, especially if we maintain some vigilance about the inherent difficulties in MT evaluation.

7. Conclusion

We have examined in this chapter why it is that evaluation takes such a pre-eminent role in the field of MT. It has been a highly visible aspect of the promise and failings

of MT since its beginnings. It is profoundly important, because of the investment that goes both into development of MT approaches, and adoption of MT processes in a work environment. And evaluation is very difficult, because translation does not generally permit comparison against single standards, and because the variety of uses and users require particular investigations of particular properties of MT systems.

We find that we are quite dependent on the intuitive, subjective judgments of people — sometimes monolingual, relatively disinterested speakers of the target language, sometimes expert translators, sometimes people with other expertise who have or will have a vested interest in making MT work for them. In some sense we despair of this reliance, because it appears so difficult to quantify, and to extract useful judgments about, say, the fidelity and intelligibility of MT output, from the endless variability of people, times, and circumstances. But we have discussed in this chapter ways in which the variables can be addressed, allowing us to capture the consistent judgments that enable us to tell which theories, approaches, and systems are more amenable to particular needs.

The different types of evaluation we described here are intended to tell us what we need to know about MT systems at different points in the system's life, and from the perspectives of the people who must use it or make decisions about it. Perhaps if MT systems generally produced nearly perfect output, we would have fewer of these types — I imagine that spell-checkers can use the same methods for feasibility, internal, and declarative evaluations. But MT is not nearly perfect, and indeed we have clearly seen that what "perfect" might mean in MT is relative to a number of attributes, everything from linguistic coverage to operational cost.

One of the things we might say in common about all of the evaluation types is that their methods must be designed and done carefully, to control for the sources of variance. Most of the types take time, effort, and coordination to perform. Some way to automate some or all of the evaluation types would be extremely beneficial for the field, allowing for the critical choices of all the stakeholders to be made much more rapidly and consistently. For some types, e.g., usability, automated measurement may be possible today. For declarative and internal evaluations, automation is much harder because of the "ground truth" problem that translation has. A solution may lie in discovering consistent correlations between the attributes we need to measure and measurements we can make automatically.

Whatever new methods may emerge, and whatever methods will ultimately be unnecessary, it is clear that evaluation will remain very near the center of MT awareness.

8. Further reading

The most comprehensive coverage of evaluation issues and techniques remain, remarkably enough, two very old works. One is the ALPAC report we discussed early in the chapter (ALPAC, 1966) and the other is *Critical Methods for Evaluating the Quality of Machine Translation* by Georges van Slype (1979). Both of these works provide an array of techniques which, though the context is quite dated, still provide useful measures and rationales for them. A much more recent, but almost equally useful treatment of evaluation as it relates to the users and uses of MT, is Arnold et al. (1993), from whom the treatment in this chapter derives much of its structure. Additionally the cited reference to the EAGLES evaluation standards, along with a more recent, joint US–European effort called the International Standards for Language Engineering (ISLE) are evolving sources of thought on the organising principles for the coverage of evaluation methods over specific MT attributes.

Notes

1. See Henisz-Dostert et al. (1979); Hutchins and Somers (1992); Hutchins (1997).
2. See also Hutchins (1996) for a useful summary of the ALPAC Report's findings.
3. For reasons which will become apparent, we do not attempt a gloss of this example.
4. See, for example, Campbell and Stanley (1963).
5. This characterization of types is largely based on the work of Arnold et al. (1993), augmented by the models of van Slype (1979) and Vasconcellos (1992).
6. Key: S – sentence, VP – verb phrase, NP – noun phrase, V[COP] – copular verb, ADJ – adjective, V[PPART] – verb past participle.
7. Examples of approaches to addressing these issues are described in the ALPAC (1966) report, Lange and Gerber (1994), and Minnis (1993).
8. See, for example, Sinaiko (1979) or Taylor and White (1998).
9. See for example Dostert (1973), Jordan et al. (1993).
10. Figures 5 and 6 are taken from Nomura and Isahara (1992).
11. For example, OVUM (1995).
12. Cf. Hirschman et al. (2000).

References

- ALPAC (1966) *Language and Machines: Computers in Translation and Linguistics*, Report by the Automatic Language Processing Advisory Committee (ALPAC), Publication 1416, National Academy of Sciences National Research Council, Washington, D. C.
- ARPA (Advanced Research Projects Agency) (1993) *Human Language Technology: Proceedings of a Workshop*. San Francisco: Morgan Kaufmann.
- Arnold, Doug, Louisa Sadler, and R. Lee Humphreys (1993) "Evaluation: an Assessment", *Machine Translation* 8, 1–24.
- Campbell, D. T. and J. C. Stanley (1963) *Experimental and Quasi-Experimental Designs for Research*. Skokie, Ill.: Rand McNally.
- Dostert, B. (1973) *User's Evaluation of Machine Translation, Georgetown MT System, 1963–1973*, Rome Air Development Center Report AD-768 451, Texas A&M University.
- EAGLES (Expert Advisory Groups in Language Engineering Standards) (1996) *EAGLES Evaluation of Natural Language Processing Systems*, Center for Sprogteknologi, Copenhagen, Denmark.
- ELRA (European Language Resources Association) (1998) *First International Conference on Language Resources and Evaluation (LREC 1998): Proceedings*, Granada, Spain.
- Farwell, David, Laurie Gerber and Eduard Hovy (eds) *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas AMTA '98*. Berlin: Springer.
- Flanagan, Mary A. (1994) "Error Classification for MT Evaluation", in *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, Maryland, pages 65–71.
- Heinisz-Dostert, Bozena, R. Ross Macdonald and Michael Zarechnak (1979) *Machine Translation*. The Hague: Mouton.
- Helmreich, Stephen and David Farwell (1998) "Translation Differences and Pragmatics-based MT", *Machine Translation* 13, 17–39.
- Hirschman, Lynette (1998) "Language Understanding Evaluations: Lessons Learned from MUC and ATIS", in ELRA (1998), pp. 117–122.
- Hirschman, L., F. Reeder, J. Burger and K. Miller (2000) "Name Translation as a Machine Translation Evaluation Task", in *LREC-2000: Second International Conference on Language Resources and Evaluation, Workshop Proceedings: Evaluation of Machine Translation*, Athens, pp. 21–28.
- Hutchins, W. John (1996) "ALPAC: The (In)Famous Report", *MT News International* 14 (June 1996), 9–12.
- Hutchins, W. John (1997) "From First Conception to First Demonstration: The Nascent Years of Machine Translation, 1947–1954. A Chronology", *Machine Translation* 12, 195–252.
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*. London: Academic Press.
- JEIDA (Japanese Electronic Industry Development Association) (1992) *JEIDA Methodology and Criteria on Machine Translation Evaluation*. Tokyo: JEIDA.
- Jordan, Pamela W., Bonnie J. Dorr and John W. Benoit (1993) "A First-Pass Approach for

- Evaluating Machine Translation Systems”, *Machine Translation* 8, 49–58.
- Lange, Elke and Laurie Gerber (1992) “Internal Evaluation: Quality Analysis, an Internal Evaluation Tool at SYSTRAN”, in *MT Evaluation: Basis for Future Directions. Proceedings of a workshop sponsored by the National Science Foundation*, San Diego, California, pages 55–56.
- Minnis, Stephen (1993) “Constructive Machine Translation Evaluation”, *Machine Translation* 8, 67–76.
- Nagao, Makoto, Jun-ichi Tsujii and Jun-ichi Nakamura (1985) “The Japanese Government Project for Machine Translation”, *Computational Linguistics* 11, 91–109.
- Nomura, Hirosato and Hitoshi Isahara (1992) “JEIDA’s Criteria on Machine Translation Evaluation”, in *International Symposium on Natural Language Understanding and AI (NLU & AI) as a part of International Symposia on Information Sciences (ISKIT’92)*, Iizuka, Fukuoka, Japan, pages 107–117.
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2001) *Bleu: a Method for Automatic Evaluation of Machine Translation*, IBM Research Technical Report, Thomas J. Watson Research Center, Yorktown Heights, NY. [available on the Internet]
- Sinaiko, Harold W. (1979) “Measurement of Usefulness by Performance Test”, in van Slype (1979), page 91.
- Taylor, Kathryn and John S. White (1998) “Predicting What MT is Good for: User Judgments and Task Performance”, in Farwell et al. (1998), pages 364–374.
- van Slype, Georges (1979) *Critical Methods for Evaluating the Quality of Machine Translation*, Report BR 19142 prepared for the European Commission Directorate General Scientific and Technical Information and Information Management, Bureau Marcel van Dijk, Bruxelles.
- Vasconcellos, Muriel (1992) “Panel: Apples, Oranges, or Kiwis? Criteria for the Comparison of MT systems”, in *MT Evaluation: Basis for Future Directions. Proceedings of a workshop sponsored by the National Science Foundation, San Diego, California*, pages 37–50.
- White, John S. (1995) “Approaches to Black Box MT Evaluation”, in *MT Summit V Proceedings*, Luxembourg, [no page numbers].
- White, John S., Theresa O’Connell and Francis O’Mara (1994) “The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Future Approaches”, in *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, Columbia, Maryland, pages 193–205.
- White, John S. and Theresa A. O’Connell (1996) “Adaptation of the DARPA Machine Translation Evaluation Paradigm to End-to-End Systems”, in *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal, Quebec, pages 106–114.

CHAPTER 14

Controlled language for authoring and translation*

Eric Nyberg, Teruko Mitamura and Willem-Olaf Huijsen
Carnegie Mellon University, Pittsburgh, PA / University of Utrecht,
Netherlands

1. Introduction

Both humans and computers may experience difficulty in understanding and translating natural language, due to its inherent ambiguity and complexity. Controlled languages (CLs) address this problem by defining guidelines for and restrictions on the language which is used to author texts. Through the use of CL, texts become easier to read and understand. This in turn enhances the efficiency and accuracy of the tasks associated with technical documentation, and improves the quality of human- and machine-translated text. CLs are typically applied to different types of technical documentation, such as operating instructions, installation and maintenance manuals, etc.

The remainder of this section presents some necessary background, including definitions, a comparison of different approaches, and a survey of some current and historical CL systems. In Section 2, we address the issue of how CL is applied to a particular writing process through document checking and correction. In Section 3, we discuss various characteristics of CL when it is used in conjunction with machine translation (MT). A discussion of evaluation methods is presented in Section 4. A detailed case study of a particular CL system (*KANT*) is presented in Section 5. Section 6 concludes with a summary plus a discussion of future trends in CL research and development.

1.1 What is controlled language?

A CL is an explicitly defined restriction of a natural language that specifies constraints on lexicon, grammar, and style. It is important to note that there is no single CL, say for English, which is approved by some global authority. In practice, there are several

different definitions of CL, which are proposed by individual groups of users or organizations for different types of documents. CL can be used solely as a guideline for authoring, with self-imposed conformance on the part of the writer; CL can be used with software which performs a complete check of each new text to verify conformance; and CL can also be incorporated into a system for automatic MT of technical text. In all cases, the overall aim is to reduce the ambiguity and complexity of the text, whether it is processed by machine or read by humans only.

A common goal of CL is a reduction in the number of words that may be used, and an adherence to the principle of **one-to-one correspondence** between word forms and concepts. This principle rules out cases where a word form corresponds to more than one concept (homonymy and conversion¹) and cases where a concept can be expressed by more than one word form (synonymy and spelling variants). A CL lexicon typically includes both words that are **approved** and words that are not approved (“**unapproved**”) for use in writing texts. For approved words, it may include the orthographical form, the syntactic category, a definition, and one or more examples of their use. For unapproved words, it may include the orthographical form, the syntactic category, a definition, and one or more suggestions for approved words that may be used to express the same meaning. Consider the examples in Figure 1.

For machine-oriented CLs (see Section 1.2), each lexical entry may include many other pieces of information required for the computational processing of the text or for terminology management. For example, lexical entries might include more detailed information on the term’s syntactic properties, semantic categories, and date of creation or latest modification.

Approved word	<i>prevent</i> (v)
Definition	To make sure that something does not occur
Example	<i>Attach the hoses to the fuselage to prevent their movement.</i>
Unapproved word	<i>preventive</i> (adj)
Approved alternative	<i>prevent</i> (v)
Unapproved example	<i>This is a corrosion preventive measure.</i>
Approved rewrite	<i>This prevents corrosion.</i>
Approved word	<i>right</i> (adj)
Definition	On the east side when you look north.
Example	<i>Do a flow check of the pump in the right wing tank.</i>
Unapproved word	<i>right-hand</i> (adj)
Approved alternative	<i>right</i> (adj)
Unapproved example	<i>The fuel connector is in the right-hand wing.</i>
Approved rewrite	<i>The fuel connector is in the right wing.</i>

Figure 1. Examples of Simplified English: *prevent* vs. *preventive* and *right* vs. *right-hand*.²

Another basic goal of CL is to reduce or eliminate the use of ambiguous and complex sentence structures. Consider the following typical writing rules of Simplified English, an example of a human-oriented CL (see Section 1.2):

- “Do not use sentences with more than 20 words.”
- “Do not use the passive voice.”
- “Do not make noun clusters of more than four nouns.”
- “Write only one instruction per sentence.”
- “Make your instructions as specific as possible.”
- “Use a bulleted layout for long lists.”

We will see further examples of writing rules in our discussion of the PACE CL in Section 3.1. Grammar rules for machine-oriented CLs are usually more specific than those for human-oriented CLs, as we will see in more detail in our discussion of KANT in Section 5.

1.2 Human-oriented and machine-oriented CLs

CLs can be characterized as human-oriented or machine-oriented. **Human-oriented** CLs intend to improve text comprehension by humans; **machine-oriented** CLs intend to improve “text comprehension” by computers. Although these two orientations have a lot in common (many simplifications are likely to increase both human and computer comprehension), there are also a number of important differences. Examples of restrictions on writing that aid both humans and computers are the limitation of sentence length and the obligatory use of commas between conjoined sentences. Other rules aid human comprehension more than computer comprehension: for example, “dependent clauses that express a condition on the action in the main clause must precede the main clause” (this order is easier for humans to understand, but does not make a sentence easier for the computer to process). Conversely, there are writing rules that are of greater benefit to computational processing: for example, a restriction on the use of pronouns (humans do not have much difficulty understanding what pronouns refer to, but resolving such references can be difficult for the computer). A general difference is that writing rules for the machine-oriented CLs must be precise and computationally tractable; for example, “Do not use sentences of more than 20 words”. Writing rules which are effective for human-oriented CLs may be computationally intractible, or intentionally vague; for example, “Make your instructions as specific as possible”, or “Present new and complex information slowly and carefully”.

In practice, however, it is often difficult to classify a CL as either human-oriented or machine-oriented, since often simplification works both ways.

1.3 Advantages and disadvantages of CLs

The general advantage of CLs is that they make many aspects of text manipulation easier for both humans and computer programs. The reduction in homonymy, synonymy, and complexity of the lexicon and the adherence to writing rules may improve the **readability** and **comprehensibility** of the text. Consequently, the performance of tasks that involve the documentation can be more efficient and effective. This advantage is especially relevant for complex texts, and also for non-native speakers. All documents written in the CL will exhibit a **uniformity** in word choice, use of terminology, sentence structure, and style, which makes them easier to maintain and reuse.

For organizations that have complex technical manuals for high-precision tasks, clearer documentation reduces the chance of misunderstanding, so that there is less chance for accidents, poor resource utilization, and other liability risks.

It is also the case that the use of CL improves both the **consistency** and **reusability** of the source text. By encouraging authors to use standard terminology and sentence structures, a uniformity of style is achieved which allows text written for one manual or product to be reused elsewhere when appropriate. It is also the case that the use of translation aids (such as translation memory tools) become more effective when the source text is more consistent and less varied. When deployed with care, a CL can dramatically increase the reusability of source text, which reduces the overall cost of authoring new documentation. CL used in conjunction with translation memory can also improve the percentage of previous translations which are reusable, thus lowering overall translations costs.

On the computational side, the use of a CL can also make the computational processing of text more efficient and effective. Depending on the specific CL and the specific computational task, it may even be feasible to prove that the computational processing of text will succeed. CL has received significant attention from the research and development community in the field of MT, since the use of CL restrictions usually improves the translatability of technical text (leading to higher-quality translations).

The use of CLs also has a number of potential drawbacks however. From the author's point of view, the writing task may become more time-consuming. It can take more concentration to write documents if they must conform to the rules of a CL, which can slow down the writing process. CLs which are not supported by automatic checking require self-vigilance on the part of the author, which can also be time-consuming. Rewriting a sentence which does not conform is often more complex than the simple substitution of approved counterparts for unapproved words, and sometimes requires rewriting the whole sentence. Consider the follow-

ing case in AECMA Simplified English (SE). The use of the phrase *according to* is unapproved; one is advised to use the verb *refer to* instead. Thus, SE disapproves of (1a) which could be rewritten to SE as (1b).

- (1) a. Calibrate test set according to manufacturer's instructions.
b. To calibrate the test set, refer to the manufacturer's instructions.

In addition to the writing task becoming more complex, authors may also experience a reduction in the power of expression if words that express the meaning they want to convey are unapproved and no good alternatives are provided. It has been claimed that writing in a CL takes up to 20% longer (Goyvaerts, 1996: 139).

Because of these issues, the introduction of CL in an organization may meet resistance from technical authors and translators. Authors and translators may feel that their writing skills are severely limited by the CL. Authors and translators should therefore be involved in all stages of CL creation and deployment, so that they have the opportunity to give their input into the **language definition process**, as well as participating in the introduction and evaluation of the finished project.

When integrating a CL into an existing **document production process**, it is important to consider the impact, if any, on the existing process. In particular, the addition of an explicit **verification** or checking phase must be accounted for. Although this typically adds more time to the authoring process, it generally reduces the amount of revision required at the editorial level before documents are approved for publication.

On the financial side, the introduction of a CL can involve a substantial investment. An organization can either license and customize an existing CL product, or bear the expense of designing, developing, and maintaining their own CL. Defining a new CL involves several phases of linguistic analysis and terminology development.³ In addition, development may include the in-house construction or purchase of a CL checker (see Section 2). The CL must also be maintained: it must continuously adapt to changing needs and wishes, new terminology and new standards, etc.

For human-oriented CLs, there is yet another difficulty: it is hard to determine the effects of their use, and, also, there have been but a few studies on this subject. The evaluation of CL systems will be discussed at greater length in Section 4.

Despite the additional costs of introducing CL into an existing document production process, the long-term advantages typically outweigh the costs for organizations which produce a high volume of documentation per year, and for whom the gains in consistency, reusability, and translatability are highly significant. In the following section, we present a survey of some existing CL systems.

1.4 A survey of CLs

Before we begin our survey of CLs, it is important to note that most CL standards are considered proprietary by the organizations that develop them. Consequently, it is often hard or even impossible to obtain detailed information on their definition.⁴

The notion of CL can be traced back to the work of Charles K. Ogden in the 1930s and 1940s (Ogden, 1932, 1942). His “Basic English” consists of 850 words and a few rules that describe how to inflect these words and how to derive other words. This simplified language was intended to be used both as an international language and as a foundation for learning standard English. However, at that time it was seen as a mere curiosity, unsuitable for any practical purpose, so that Basic English has never been widely used.

The first CL put to actual use was Caterpillar Fundamental English (CFE), used by Caterpillar Inc. in the 1970s (see Section 5.1 for more details). CFE inspired other CLs such as Smart’s Plain English Program (PEP), E. N. White’s International Language of Service and Maintenance (ILSAM), J. I. Case’s Clear and Simple English (CASE), and PACE (see below). PEP in turn gave birth to CLs used by Clark, Rockwell International, and Hyster (Hyster’s Easy Language Program, HELP), while ILSAM can be considered the root of the CLs of AECMA (SE), IBM, Rank Xerox, and Ericsson Telecommunications.

In 1979, the Douglas Aircraft Company constructed a dictionary of about 2,000 words which it uses for technical manuals. In the UK, Perkins Engines Ltd. introduced Perkins Approved Clear English (PACE) to simplify their publications and to aid translation in the 1980s (discussed below). At Wolfson College in Cambridge, three human-oriented CLs were developed for fast and accurate communication: Airspeak for air-traffic control, Seaspeak for maritime communication, and Policespeak for the English and French police in the Channel Tunnel.

As a part of a major modernization effort by Caterpillar Inc., CFE has been replaced by Caterpillar Technical English (CTE). The large volume of documentation (over 100,000 new pages each year) and the requirement of translation in up to 35 languages necessitate heavy automation of the translation process. To this end, Caterpillar engaged Carnegie Group Inc.⁵ and the Center for Machine Translation (CMT) at Carnegie Mellon University to develop and deploy a combined authoring and translation system based on CMT’s KANT technology. The KANT MT system⁶ produces high-quality translation provided that the source language is strictly controlled for both vocabulary and grammar. We will discuss the CTE/KANT system in more detail in Section 5.

Carnegie Group Inc. was also engaged by Diebold Inc., a global leader in card-based transaction systems and security, to develop a CL called Diebold Controlled English. In Sweden, the Scania company, a leading manufacturer of heavy trucks,

has defined ScaniaSwedish, a CL for the automatic translation of truck maintenance documentation. Other CLs (mostly machine-oriented) are General Motors' Controlled Automotive Service Language, Controlled English at Alcatel Telecom in Belgium, Siemens Dokumentationsdeutsch in Germany, and GIFAS Rationalized French for the French aerospace industry.

A prime example of fundamental research into CLs is Attempto Controlled English (ACE). At the University of Zurich, this language is being developed for the formal specification of software. Using ACE, specifications can be written in (controlled) natural language, so that they are readily understood, while their interpretation is unambiguous.

ACE [...] is designed so that a text can be represented unambiguously in first-order predicate logic. The translated document can be verified for completeness and consistency by querying it in ACE [...] Thus validation and prototyping in concepts close to the application domain become possible and the results can be understood by all parties concerned. (Fuchs and Schwitter, 1996).

One of the best-known CLs is SE, already mentioned, a human-oriented CL for aircraft-maintenance documentation. In 1979, the Association of European Airlines asked the European Association of Aerospace Industries (AECMA) to investigate the readability of maintenance documentation in the civilian aircraft industry. In the following years, SE was developed: maintenance documentation was analysed and a basic vocabulary and a set of writing rules were set up. SE is described in a document known as "the SE Guide" first released in 1986. By now, it is a world-wide standard for aircraft-maintenance documentation, and an example for other fields.

The SE Guide includes a limited basic vocabulary (about 3,100 words) and a set of 57 writing rules. The guiding principle in the SE lexicon, as in any other CL lexicon, is "one word one meaning". For example, the verb *fall* may be used only with meaning 'to move down by the force of gravity', and not with meaning 'to decrease'. The vocabulary can be extended with aircraft-industry terminology as needed: technical names, which are nouns denoting specialized aircraft entities (e.g. *fuselage*, *air-traffic control*), and manufacturing processes, which are verbs denoting industrial activities (e.g. *anneal*, *polish*). The writing rules pertain to punctuation, word choice, sentence length, syntactic constructions, text structure, style, and layout.

A few studies on the effect of the use of SE are reviewed in Section 4.

2. CL checking and correction

A CL checker is a specialized piece of software which aids an author in determining whether a text conforms to a particular CL. Checker programs verify that all words

are approved and that the writing rules are obeyed. In addition, they may offer help to the author when words or sentences not in the CL are found during checking. We will now go into more detail on both the checking and the correction of CL, and then we present an inventory of existing CL checkers.

2.1 CL checking

CL checkers are programs which assist authors in determining whether their text complies with the specification of a CL. This assistance is generally given as a series of critiques or issues that are raised with respect to the text, communicated to the user as text messages by the software. Important quality measures include the percentage of critiques which are appropriate (precision), and the percentage of potential critiques which are found by the checker (recall). Both precision and recall should be kept as high as possible.

Checking compliance with the CL lexicon consists largely of well-understood computational processes such as determining the syntactic category of words in their context, morphological analysis, and looking up words in the lexicon. It is much more difficult for a checker to determine the meaning of a word given its usage in a given text. Determining the meaning of a word is necessary because many words in the CL lexicon are either approved or unapproved depending on their meaning. For example, in everyday English, the adjective *right* may mean either ‘the opposite of left’, or ‘correct’. In SE, the word *right* is approved only if it is used with the former reading. If it is used with the latter reading, it is unapproved, and should be replaced with the approved word *correct*.

Given the current state of the art in computational linguistics, there is a great variation in the degree to which individual writing rules can be checked automatically. Some writing rules are easy to check (e.g. “Do not use sentences with more than 20 words”) while other rules are harder to check (e.g. “If possible, use an article before a noun phrase”), and yet others are impossible to check automatically (e.g. “Make your instructions as specific as possible”).

The ability to check compliance with writing rules adequately depends largely on the depth of syntactic, semantic, and pragmatic analysis which can be done automatically. In general, the writing rules in a CL can either be **proscriptive**, explicitly describing structures which are not allowed, or **prescriptive**, explicitly describing those structures which *are* allowed.

The prescriptive approach is generally implemented using heuristic patterns or templates which are matched against the input to detect structures which are not allowed. This approach typically requires less work, because it only needs to focus on finding unacceptable sentences, and does not need to specify allowable structures exhaustively. However, the prescriptive approach can overlook certain problems,

and is more likely to give inappropriate feedback (for example, when a general pattern is matched by an exceptional sentence which is perfectly acceptable).

The prescriptive approach is more labor-intensive, since it requires a definition of each and every linguistic structure that is allowable in the CL. If the prescriptive grammar is implemented in a computational system, then each sentence can be parsed to see if it conforms to the rules of the grammar (and is hence an allowable sentence). Because of the thoroughness of the analysis, this approach is less likely to give inappropriate feedback. On the other hand, there are likely to be some sentence structures that are overlooked in the original language definition but still considered necessary; hence the grammar rules must undergo extension and tuning during initial use.

2.2 CL correction

In addition to pointing out violations of the CL, a checker may also offer help in the form of proposed corrections. This may consist of general advice on making text conform to the writing rules, specific propositions for correction to be selected and confirmed by the author, or even provision of fully automatic correction. Because of the potentially tedious and repetitive nature of the checking and correction process, any automation is likely to be welcomed by the authors, provided that there are few false alarms, and the authors retain final approval of the corrections proposed by the system. This is facilitated by making correction interactive, and by giving the authors the possibility to ignore potential errors and proposed corrections. Correction on the lexical level is mostly unproblematic if the unapproved word and its approved counterpart have the same syntactic properties, and they are near-synonyms. If they differ in these aspects, it is more difficult to do automatic correction, since this may require restructuring of the sentence. If the words are not near-synonyms, automatic correction is even harder, or even impossible. Resolving such situations usually requires human judgement.

Correction on the level of writing rules is more difficult. Analogous to the situation for analysis, there is a heuristic approach, and a more principled approach. The heuristic approach uses pattern substitution methods to correct violations. A more principled approach is known as **correction as translation**. It formulates correction as a translation problem, so that MT technology can be applied:⁷ correction is translation from the unrestricted language to the CL. This approach requires full computational grammars of both the unrestricted language (the source language) and the CL (the target language), and also a — preferably unambiguous — mapping from words and grammar rules of the source language to their CL counterparts.

2.3 Survey of CL checkers

Most existing CL checkers have been developed in-house. Some companies offer checkers commercially. First, we enumerate a number of in-house checkers. For example, Perkins Engines Ltd. has developed a checker for PACE,⁸ and a vocabulary checker is planned for ScaniaSwedish. For Alcatel Telecom's Controlled English, a checker has been developed in the CEC-funded SECC project ('A Simplified English Grammar and Style Checker/Corrector'), by participants Siemens Nixdorf, the University of Leuven, and Cap Gemini Innovation. This software checks technical English documentation in the field of telecommunication. The SECC checker is based on the *Metal MT* system: checking is conceived as translation from English to the CL. Given a single sentence as its input, the system outputs the input sentence annotated with error messages, together with a suggestion for correction of the whole sentence. The commercialization of this checker was foreseen right from the start.

For AECMA SE, there are a number of checkers. The Boeing company is required by international and contractual agreement to supply maintenance documentation in SE. In order to check compliance, the Boeing Simplified English Checker (*BSEC*) was developed, for internal use only. The checker relies on a grammar formalism inspired by Generalized Phrase Structure Grammar. Structural ambiguity is resolved with statistical methods. Currently, the checker is being enhanced with semantic and pragmatic language checking capabilities. The new checker is known as *EGSC* (Enhanced Grammar, Style, and Content Checker). Another checker for SE is *Eurocastle*, developed by the Aérospatiale Research Center at Suresnes, Paris, and GSI-Erli (now Erli) in the EUREKA-funded *GRAAL* project.

A few companies offer CL checkers commercially. GSI-Erli (now Erli) marketed a checker for SE called *AlethCL*. However, this product was discontinued in early 1997. The LANT company in Leuven, Belgium, is the vendor of the *LANTmaster* CL checker, based on the *Metal MT* system and the experience of the SECC project.

Carnegie Group, Inc. has developed a CL checking system called *ClearCheck*. Two applications — differing considerably in subject area, scope, and coverage — are in use at Caterpillar Inc. and Diebold Inc. Finally, Smart Communications Inc. contends with its CL checker *MAXit*.

3. CL for MT

MT is potentially one of the most interesting computational applications of CL. If a CL and an MT system are attuned to each other, MT of texts written in that CL can be much more efficient and effective, requiring far less — or ideally even no —

human intervention. Most internationally operating organizations produce both their internal documentation and their product's user manuals in a number of languages. Good technical documentation is an important factor in the overall quality of the organization's products. The use of a CL can improve the quality of the documentation and can speed up human and machine translation. Thus, costs are reduced and the foreign-language manuals are available earlier, which shortens the time-to-market of the associated products. Thus, the use of CLs can improve the competitiveness of such organizations.

An important distinction that we will make here is that between improving the quality of MT by making use of relatively loosely defined CLs, as discussed in the previous sections, and aiming at fully automatic, high-quality translation for strictly defined CLs.

3.1 MT for loosely defined CLs

Most of the CLs used in combination with MT that we have discussed above aim at improving the quality of the source-language text in order to make subsequent translation by humans and/or machines easier. Since their specification is often not very precise, we will call them **loosely defined CLs**.

A success story in this context is that of Perkins Engines Ltd. This company is a leading world-wide manufacturer of diesel and other engines. The frequent introduction of new products and modification of existing products calls for the rapid production of documentation in five languages: English, French, German, Spanish, and Italian. To simplify English publications for non-native speakers, and to aid translation, be it conventional or computer-assisted, Perkins introduced PACE in 1980. PACE consists of a lexicon of approximately 2,500 words, together with a set of ten writing rules (Pym, 1990: 85f):

1. Keep sentences short.
2. Omit redundant words.
3. Order the parts of the sentence logically.
4. Do not change constructions in mid-sentence.
5. Take care with the logic of 'and' and 'or'.
6. Avoid elliptical constructions.
7. Do not omit conjunctions or relatives.
8. Adhere to the PACE dictionary.
9. Avoid strings of nouns.
10. Do not use 'ing' unless the word appears thus in the PACE dictionary.

Between 1984 and 1987, a computer-assisted translation system, Weidner's *Micro-Cat*, was introduced to speed up the production of documentation written in

PACE. As for the benefits of using PACE and *MicroCat*, Pym reports that

...text written in accordance with PACE provides a good source text in natural and clear English. This leads to good raw translation, which often requires very little post-editing. It would appear that not all translators like post-editing, but those that do, gain satisfaction from their rate of output and are fascinated with what the system can do [...] The rate of post-editing is three to four times faster than for conventional translation. (Pym, 1990: 91).

By using a combination of CL with MT, Perkins obtained an ensured consistency of the use of terminology, and greatly reduced translation time and translation costs, leading to a higher quality of the documentation.

3.2 MT for strictly defined CLs

In contrast to a loosely defined CL, a strictly defined CL is a CL with a formally specified syntax. Such a language is an interesting point of departure for automatic translation: by choosing the restrictions imposed by the CL appropriately, it may be possible to guarantee fully automatic, high-quality translation for texts that adhere to it.

An example of a strictly CL is the work at Cap Gemini's Lingware Services.⁹ Lingware Services aims at building systems for fully automatic, high-quality MT of texts written in CL. Such systems are assumed to consist of two main modules: a word processor enhanced for CL authoring, and a translation module. The author may use the word processor to enter the source text; supporting functions can then be used to ensure that the text satisfies the restrictions of the CL. The CL itself is to be designed in such a way that user involvement is limited to the phase of document creation: subsequent translation should fully automatically produce grammatically correct target-language expressions that are acceptable as translations and that require no (or, at worst, minimal) post-editing.

Since the early 1990s, Lingware Services have developed software to support large-scale document creation and translation using CLs for on-line help texts, software manuals, and aerospace maintenance manuals. Their systems include the automatic correction of morphosyntactic, terminological, and spelling errors, and the generation of warnings for errors that cannot be corrected automatically. Their authoring tools provide support for the lexical, grammatical, and style specifications of the CL.

The KANT system is an example of an MT system for strictly defined CLs which has been deployed in industry. A detailed case study, including further detail regarding machine translation for strictly CLs, is presented in Section 5.

4. Evaluation of CLs

The central claim about the use of human-oriented CLs is that they improve readability and comprehensibility of text. The fact that SE is in wide use in the aircraft industry seems to underline their practical relevance. However, the effects of the use of CL are difficult to determine, and there are but a few empirical studies on this subject. We will first discuss what makes this issue so difficult, and will then sum up four of the available studies.

For human-oriented CLs, the assessment of the effect of the use of a CL is a difficult issue. Adequate evaluation must take into account a large number of variables, such as:

- the number of texts and test persons used in the evaluation,
- the amount of time available to the test persons to execute the test,
- the complexity of the texts and their subject matter,
- the degree to which the test persons are familiar with the subject matter and the CL,
- whether they prefer the CL texts to the uncontrolled ones,
- in how far they are more inclined to use the texts,
- and whether they are native speakers or not.

Many of these variables are hard to quantify. Consequently, the effect of the use of a CL is hard to determine. Also, as most CLs are specified semi-formally, it is often difficult to check conformance. Due to the semi-formal or even informal way in which many CLs are defined, it is often unclear which part of the definition of the CL should be applied to determine conformance. Some writing rules are vague, for example “Keep sentences short”. Others may seem to contradict each other. Consider, for example, “Keep sentences short” and “Avoid strings of nouns”. Adherence to the latter would suggest rewriting the 7-word phrase in (2a) as the 11-word phrase (2b), which clashes with the former writing rule.

- (2)
 - a. the nose landing gear uplock attachment bolt
 - b. the bolt that attaches the uplock to the nose landing gear

A further difficulty in establishing the effect of the use of CL is that it is unclear what the contribution of each individual writing rule is to the overall effect of the CL. Some writing rules may do more harm than good.

For example, the second writing rule of PACE reads “Omit redundant words”. Without an unambiguous statement about what should be considered redundant, this rule may inadvertently lead to the omission of words that are crucial to a clear understanding of the text. Note that SE has a rule that apparently advocates the contrary, saying “Do not omit words to make your sentences shorter”.

Much empirical and theoretical work remains to be done here.

Nevertheless, there have been a few empirical studies on the effects of the use of CL that support the common belief that it improves **readability** and **comprehensibility**. We now sum up four of these studies, all of which concern SE. One study (Chervak et al., 1996) compared comprehension of SE and non-SE versions of workcards by 175 aircraft-maintenance technicians. It found that the use of SE gave a significant improvement in comprehension. This effect was most pronounced for more difficult workcards, and with non-native speakers of English. Another study of SE suggested that

...using SE significantly improves the comprehensibility of more complex documents. Further, readers of more complex SE documents can more easily locate and identify information within the document. (Shubert, 1995)

Concerning comprehensibility and content location scores, non-native speakers seem to benefit more from SE than native speakers. A recent study (Kincaid, 1997) is in keeping with these findings. This study shows that non-native speakers of English showed an increase of 18% on comprehension scores, as compared to standard English. A fourth study of SE (Holmbäck et al., 1996) also concludes that it can significantly improve comprehension, and that it can somewhat improve the ease and quality of human translation. The authors call for more empirical studies on SE and other CLs, giving some recommendations for such studies.

Another interesting study relating to the evaluation is Lehrndorfer's dissertation (Lehrndorfer, 1996). In this work, she discusses the linguistic and psycholinguistic issues in the development of a machine-oriented CL for German.

Although much work remains to be done on the evaluation of CLs before hard claims can be made, the few studies that are available support the belief that the use of human-oriented CLs improves the readability and comprehensibility, especially for complex texts and for non-native speakers. Apart from the field studies on the effect of the use of CLs, there are also more formal evaluation studies on the properties of CL checkers, most notably precision, recall, and convergence. **Precision** is the proportion of the number of correctly flagged errors to the total number of errors flagged; **recall** is the proportion of the number of correctly flagged errors to the total number of errors actually occurring; and (for automatic correction) **convergence** is the proportion of the number of automatically corrected sentences that are accepted when resubmitted to the total number of automatically corrected sentences. The ideal system, of course, should not report errors that are not there (100% precision); should not fail to report any errors (100% recall); and correction should eliminate all errors and should not introduce new errors itself (100% convergence). There are but a few studies on these properties of CL checkers.¹⁰

It is also possible to evaluate the effectiveness of a CL in the context of an MT system, by performing an evaluation of the translation output quality. If the assertion that CL improves the quality of source text is true, then we should expect that machine translation of CL texts should produce better results. The effect of CL is typically measured as a function of post-editing cost — if the use of CL results in translations which require less post-editing, then the overall increase in productivity can be measured and tracked over time. Likewise, it is possible to measure the productivity gains in human (manual) translation by comparing the time taken to translate uncontrolled text with the time required to translate the same text after it has been rewritten to conform to a CL.

5. Case study: The *KANT* MT system and CTE

Caterpillar Inc., a heavy equipment manufacturing company headquartered in Peoria IL, supports world-wide distribution of a large number of products and parts. Each Caterpillar product integrates several complex subsystems (engine, hydraulic system, drive system, implements, electrical, etc.) for which a variety of technical documents must be produced (operations and maintenance, testing and adjusting, disassembly and assembly, specifications, etc.). To support consistent, high-quality authoring and translation of these documents from English into a variety of target languages, Caterpillar uses CTE (Caterpillar Technical English), a controlled English system developed in conjunction with Carnegie Mellon University's CMT and Carnegie Group Inc.

Caterpillar's CTE has been deployed for checking and translation as one application of the *KANT* MT system. The implementation of CTE combines three kinds of constraints:

- **Constraints on the lexicon.** In order to reduce lexical ambiguity and complexity, constraints are placed on the source vocabulary.
- **Constraints on the complexity of sentences.** To limit parsing complexity during source analysis, the types of input sentences are limited to those necessary for concise technical authoring.
- **Use of Standardized Generalized Markup Language (SGML).** The use of SGML supports the definition of complicated domain terminology and phrasal constructions without increasing the ambiguity of the analysis grammar.

Once the CL has been defined and the data files constructed, the language may be embedded into a system for on-line document authoring which supports the following activities:

- **Vocabulary checking.** The input text is checked to ensure that it conforms to constraints on vocabulary; otherwise, the system helps the author to select alternative vocabulary.
- **Grammar checking.** The input text is checked to ensure that it conforms to constraints on grammar; otherwise, the system prompts the author to rewrite the sentence under consideration.
- **Interactive disambiguation.** If ambiguities arise during grammar checking, the system may ask the author to choose among competing analyses, encoding those choices for later use during translation.

In the remainder of this section, we discuss the history and development of CTE, the implementation of CTE in the *KANT* system, and the benefits and challenges we encountered in the development and deployment of CTE for daily use. Figure 2 shows an overview of the integrated system.

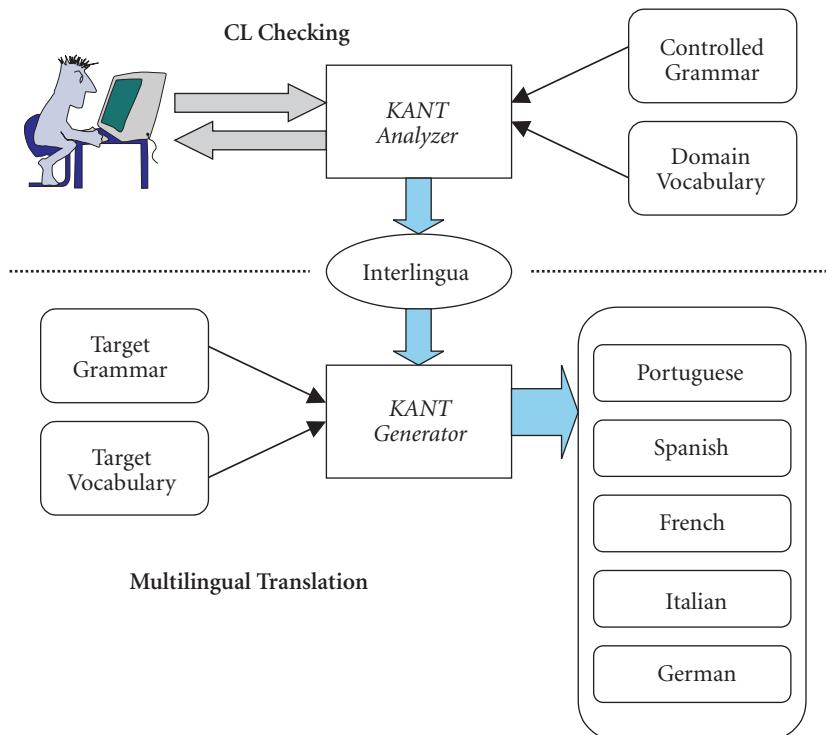


Figure 2. CL Checking and Translation in *KANT*.

5.1 A precursor to CTE: Caterpillar Fundamental English

CTE was not the first controlled English deployed at Caterpillar. In the 1970s, Caterpillar utilized a different controlled English approach called Caterpillar Fundamental English (CFE). CFE involved the use of an extremely limited vocabulary and grammar, and was intended as a form of English as a Second Language (ESL). CFE was intended for use by non-English speakers, who would be able to read service manuals written in CFE after some basic training. The CFE approach was intended to eliminate the need to translate service manuals.

CFE was designed around the basic English sentence patterns that could be learned in an introductory ESL class. The initial version of CFE, designed in 1972, had a vocabulary of about 850 terms. The intent was to use an illustrated parts book, and to illustrate the service manuals heavily so that the simplified text could be followed by the service technician. Caterpillar employed CFE for slightly over ten years, before abandoning the approach for a number of key reasons. Some of the more important reasons, which were addressed in subsequent development, include the following:

- The complexity of Caterpillar's equipment was expanding rapidly, especially in the areas of high-pressure hydraulics and electronics; a limited vocabulary was simply insufficient for these areas.
- For many cultures it is a point of national and cultural pride to have service literature translated; translated material was therefore recognized as an important marketing tool.
- The basic guidelines of CFE were not enforceable in the English documents produced. While there was a simplification (and a general improvement) in the writing of the English technical authors, very few documents, especially later in the program, were compliant with the CFE guidelines. Beyond extensive editing and proofing of the English output by human editors, there was no means of enforcing compliance.

CFE was discontinued at Caterpillar in 1982. In the mid 1980s, the first word processors were introduced into the Caterpillar writing environment, and by the late 1980s the rapid development of hardware and software technology led Caterpillar to re-examine their composition and publication procedures, with a view to gaining more automation and control over authoring. Improving the quality and reducing the cost of translation was also a consideration. It was determined that an enforceable controlled English was now possible, given advances in language-processing technology. An enforceable controlled English would enable a much higher degree of compliance than was available in the 1970s. These developments led to the design of the next generation of controlled English at Caterpillar: CTE.

5.2 Development of CTE

CFE employed under 1,000 terms; many of these had broad semantic scope and it was assumed that they would be disambiguated in context by the human reader. In contrast, there are around 70,000 CTE terms, of a narrow semantic scope, which are designed to be unambiguous to both the human reader and the checking software. The design and implementation of CTE are intended to provide the following characteristic benefits of CL authoring.

- Controlled input to MT, in order to improve translation quality and reduce the cost of manual translation to minimal post-editing.
- Standard terminology use and writing style for all English technical documentation.
- A user interface which provides on-line terminology definitions and usage information to the author.

The CTE development effort was launched in November 1991. The CTE definition, authoring software, and MT system were developed in parallel. Main categories of effort included CTE development, maintenance, and training, along with development of required document-type specifications and the authoring process itself. The personnel required at Caterpillar for CTE development, pilot, and training (from 1992 to 1997) averaged about five full-time equivalent employees per year. Required personnel included linguists, pilot authors, trainers, and mentors.

5.3 Benefits from CTE

The benefits realized from CTE authoring include:

- **Increased consistency of English writing and terminology.** The on-line checking of documents helps to enforce accepted use of grammar and terminology, resulting in more consistent documents.
- **Increased ability to reuse documents and translations.** Consistent writing and terminology use allows documents to be reused across product lines, leading to increases in production efficiency for technical manuals. This consistency in authored text results in a derived benefit for translators, since translation-memory tools are more effective with standardized source text.
- **Heightened awareness of language-related issues.** The requirement to write according to a standard has brought attention to a set of issues which are essential for high-quality multilingual documentation:
 - The value of writing guidelines and terminology management (for source and target languages);

- The effort to standardize the authoring and translation processes;
- The amount of training required for effective high-quality authoring and translation;
- The high level of personnel and skills required (authors and translators, as well as terminology experts, lexicographers, and system maintainers).

5.4 Challenges

The set of challenges we faced during CTE development are probably common to any large-scale implementation of controlled technical authoring, and include the following:

- CTE terminology maintenance is an ongoing task, which includes control of terminology proliferation, removal of redundant terms, and screening of new terms requested by authors. Similar tasks are required to keep the translated terminology up-to-date.
- Maintenance of usage examples is required. Every time the CTE grammar is improved in any way, the existing usage examples must be re-validated to ensure that they are still proper CTE.
- The CTE domain is too complex for lexicographers to anticipate all the ways authors use words; hence ambiguous phenomena cannot be defined in advance, and the lexicon and grammar must be extended through successive refinements after initial deployment of the system.
- The requirement for accurate translation is a driving force in representing semantic ambiguity during CTE terminology development. Terms that do not appear to be ambiguous during superficial review turn out to have several context-specific translations in different target languages, prompting a finer-grained (ambiguous) representation in the CTE lexicon for some terms.
- Adherence to CTE principles by authors and translators is variable and sometimes difficult to enforce. Authors may use words in senses that are not approved, and sometimes authors select the wrong meaning choices for words during interactive disambiguation (both phenomena tend to degrade the translated output). Translators sometimes have difficulty accepting the short, simple sentences which are characteristic of CTE, preferring instead to rewrite large portions of the text in translation. This unnecessary post-editing offsets the productivity gains (from higher translation quality) achieved through the use of CL.

5.5 Controlled vocabulary development

A key element in controlling a source language is to restrict the authoring of texts such that only a pre-defined vocabulary is utilized. In order to define a controlled vocabulary for a particular application domain, pre-existing documents are analyzed as an initial source of vocabulary. This initial vocabulary is further refined as the domain meanings of each term are encoded, and emerging lexical classes begin to collect domain-specific closed-class items. It is inevitable that each domain will contain a set of ambiguous terms (words for which the same root–part-of-speech pair has more than one semantic assignment), so we have also designed a method for disambiguation of lexical items in the input which supports interactive disambiguation by the author.

The first step in defining a domain vocabulary is to extract as many terms as possible from pre-existing on-line documentation. In the case of CTE, about 50 megabytes of existing corpus were used to extract a domain vocabulary for heavy equipment documentation. There are three broad categories of technical vocabulary to be considered in defining a controlled English:

- **Technical phrases.** In a given domain, there are likely to be phrases for which the meaning is difficult to recover unless the phrase is stored in the lexicon as a single unit. Such phrases include noun phrases whose meaning cannot be derived compositionally, such as *oil pan* when we assume that the word *pan* has no separate domain meaning. Phrasal verb–particle constructions such as *abide by* are also easier to analyze if taken as a unit. It is also the case that large numbers of technical noun phrases which might be compositionally analyzed can be more efficient to analyze during parsing if they too are represented as single units of lexical meaning. In the case of the *KANT* application for Caterpillar, there are about 50,000 domain phrases encoded in the lexicon.
- **Technical words.** In a typical domain, there are many single symbols which have a special meaning in the domain and are not found in other kinds of text. For example, technical documentation generally contains symbols such as acronyms (e.g., *Programmable Electronic Engine Control, PEEC*) and abbreviations (e.g., *foot pounds, ft-lb*). A given domain may also require types of lexical items that are particular to that domain (for example, a class of words denoting wire colors, or a class of words denoting labels on machine controls). Each class of technical words must be identified and filled in, generally with participation from the customer’s terminology experts.
- **Technical symbols.** Any special use of numbers, numerals, units of measure, letters of the alphabet, etc. must be specified and encoded in the lexicon as well.

One important feature of the *KANT* system is that it explicitly encodes a set of

“domain meanings” for each term in the lexicon. In knowledge-based systems like *KANT*, this meaning is used to access the domain knowledge-base during source-text analysis. Even in systems that do not utilize semantic processing, encoding domain meanings during lexicon creation helps to identify potentially difficult terms for translation. When defining a controlled English for a new domain, these three steps are taken:

- **Limit meaning per word–part-of-speech pair.** Wherever possible, the lexicon should encode a single meaning (domain concept) for each word–part-of-speech pair. This helps to reduce dramatically the amount of ambiguity in the source text, which in turn reduces the complexity of source analysis by an appreciable amount.
- **Encode meanings using synonyms.** Whenever a lexical item has more than one potential meaning in the domain, first an attempt is made to “split up” the meanings by finding separate, synonymous terms to encode them. Terms which are “split” in this manner are subsequently marked in the lexicon, so that it is possible to determine for any given word whether it has an alternate meaning which is encoded by a different term in the domain. This information can be used in support of on-line vocabulary checking (cf. Section 2.1).
- **Encode truly ambiguous terms for interactive disambiguation.** When a term simply must carry more than one meaning in the domain, either because of customer requirements or because there is no synonym available for the additional meanings, these meanings must be encoded in separate lexical entries for the same word–part-of-speech pair. If more than one such entry is activated for a given lexical item during source-text analysis, then the resulting output structure will be ambiguous (there will be more than one meaning analyzed for the sentence). In this case, lexical disambiguation must be performed to narrow the meaning further to just the meaning intended by author.

In addition to restricting the meaning of domain terms, the controlled English may also pose constraints in other areas of the vocabulary as well. Aspects of vocabulary which are commonly restricted in *KANT* applications include:

- **Orthography.** Whenever possible, the spelling, capitalization, hyphenation, and use of the slash in domain terms should be consistently specified.
- **Functional words.** Rules concerning determiners (articles), pronouns, reflexives, quantifiers, and conjunctions must be specified. Wherever possible, the use of pronouns and conjunctions should be limited, since they increase the potential ambiguity of syntactic analysis. In technical domains such as heavy machinery, there may be considerable requirements for complex units of measurement for solids, liquids, electricity, force, time intervals, etc.

- **Modal verbs.** The senses of modal verbs (*can*, *should*, etc.), and their interactions with negation must be clearly specified and taught to the authors in order to increase accurate use of these words during authoring.
- **Participial forms.** The use of participial forms (such as *-ing* and *-ed*) should be restricted. For example, *-ing* should not be used in subordinate constructions such as (3a): structures like these should be rewritten to include an explicit subject.

- (3) a. When starting the engine...
b. When you start the engine...

The *-ed* form should not be used to introduce a relative clause without explicit use of a relative pronoun; reduced relative clauses such as (4a) should be rewritten to use a relative pronoun explicitly as in (4b).

- (4) a. the pumps mounted to the pump drive
b. the pumps that are mounted to the pump drive

5.6 Controlled grammar development

When analyzing a corpus of technical documents, especially those associated with assembly, use and maintenance of machinery, one finds that the range of English constructions required for effective authoring is not large. It is often preferable to adopt a set of rules for technical writing which improve and standardize the readability of texts, even if the texts are not translated. If the grammatical constraints on the source text are formally specified and satisfied during authoring, then an MT system may take advantage of the less complex, less ambiguous texts which are produced, generally leading to better-quality output.

There are two general types of grammar restrictions: those that place constraints on the formation of complex phrases in controlled English, and those that place constraints on the structure of sentences. Phrase-level constraints include the following:

- **Verb particles.** English contains many verb–particle combinations, where a verb is combined with a preposition, adverb, or other part of speech. Particles which are part of phrasal verbs are often ambiguous with prepositions, and a controlled English should limit this ambiguity by recommending that verb–particle combinations be rewritten whenever possible. This can usually be accomplished by choosing a single-word verb instead (for example, *turn on* can be rewritten using *start*).
- **Coordination of verb phrases.** Coordination of single verbs or verb phrases is not recommended for controlled English, since the arguments and modifiers of

verbs conjoined in this manner may be ambiguous. These constructions are to be authored using conjunction of full sentences; for example (5a) is rewritten as (5b).

- (5) a. Extend and retract the cylinders.
b. Extend the cylinders and retract the cylinders.
- **Conjoined prepositional phrases.** Authors are encouraged to repeat the preposition in conjoined constructions where appropriate. It is important to distinguish the scope in phrases like (6),

(6) 5 cubic meters of concrete and sand

which could mean either that amount of mixture or that amount of each material.

- **Using the determiner in noun phrases.** In full sentences, the use of determiners (*the, a*) in noun phrases is strongly recommended, since they make the referential nature of the noun they modify more precise. This in turn supports better quality translation.
- **Nominal compounding.** In general, nominal compounding is not allowed unless it is licensed by domain rules which allow specific types of nominal compounding (e.g., wire colors, component names or modifiers, etc.). This reduces the ambiguity that would result if arbitrary noun–noun compounding were allowed.
- **Quantifiers.** Words such as *all, none*, may not appear alone, and must modify a nominal head. For example, (7a) can be more precisely written as (7b) when that is the intended meaning.

(7) a. Repeat these steps until none are left.
b. Repeat these steps until no bolts are left.

Sentence-level constraints include the following:

- **Coordinate conjunction of sentences.** In controlled English, it is recommended that the two parts of a conjoined sentence be of the same type. Sentence types should not be mixed in sentential conjunction, since a conjunction of different sentence types is difficult for a source analyzer to interpret. These constructions can be rewritten by choosing two sentences of the same type.
- **Clauses introduced by subordinate conjunctions.** Both clauses in complex sentences using subordinate conjunctions must contain a subject and a verb; if the subordinate conjunction is removed, the subordinate clause should be able to stand alone as a simple sentence. Reduced clauses without subjects such as (8a) should be rewritten to include an explicit subject (8b).

- (8) a. after installing the gear...
b. after you install the gear...
- **Adjoined elliptical modifiers.** The use of ellipsis should be ruled out whenever possible in controlled English, since it introduces potential ambiguity in ellipsis resolution. However, some elliptical phrases (e.g., *if necessary*, *if equipped*) may be required. These should be explicitly specified as a closed class in controlled English, so that the source analyzer can treat them as special cases.
 - **Relative clauses.** Relative clauses can be added to independent clauses to form complex sentences. In controlled English, relative clauses should always be introduced by the relative pronouns *that* or *which*. Relative clauses contain a gapped argument which is coreferential with the element they modify. In unrestricted English, this gap can be in the subject position of the relative clause, or in the object position of the relative clause. A third type of relative clause is introduced by a complex relative expression such as *with which* or *for whom*. *KANT* Controlled English applications typically support subject relative clauses, but not object or complex relative clauses.
 - **Wh-questions.** A given controlled English application for technical documentation may or may not require support for *wh*-questions, depending on the domain. Since deriving the long-distance dependencies between *wh*-words and their original, gapped position complicates syntactic analysis, the use of *wh*-questions can be avoided by rephrasing them as direct questions. Example (9) shows a *wh*-question and the corresponding re-write.
- (9) a. Which error did the display on the control panel indicate?
A: 123 B: 456 C: None
b. Did the display on the control panel indicate an error?
A: 123 B: 456 C: None
- **Punctuation.** The rules for consistent, unambiguous use of comma, colon, semicolon, quotation marks, and parentheses as inter- and intra-sentential punctuation should be clearly stated in the controlled English specification.

5.7 Text markup

In recent years, there has been much emphasis on the use of SGML and similar generalized markup languages for document production (see Chapter 4, Section 3). The *KANT* system supports the use of SGML tagging, and in doing so takes advantage of several positive features of SGML which reduce the complexity of source-text analysis.

The use of SGML markup in controlled-English text improves the quality of both the source and target text in the following ways:

- **Formalizing document structure.** A typical SGML implementation specifies tags to be used to mark paragraphs, lists of bulleted or enumerated items, titles and headings, tables, etc. When document context is tagged with SGML, it can be used as another source of information during analysis.
- **Limit complexity of analyzing domain vocabulary.** When SGML is used to identify items that fall into the same semantic class (e.g., part numbers, serial numbers, model names), these items need not be explicitly represented in the lexicon, allowing significant reduction in the size of the lexicon in a large technical domain with lots of component identifiers.
- **Reduce lexical ambiguity.** Sequences of symbols such as integers or alphanumeric, which might be ambiguous when untagged, are unambiguous when tagged.
- **Simplify analysis of domain-specific constructions.** When a technical domain requires that complicated sequences of numeric identifiers, modifiers, and component names be analyzed as noun phrases, the use of SGML tags can dramatically reduce the complexity of source analysis. Instead of allowing arbitrary composition of numbers and modifiers using unrestricted, recursive grammar rules, specific sequences of tagged elements may be introduced as right-hand-sides of grammar rules.

The following are some examples of SGML tagging conventions which improve the quality of the source text and should be considered for controlled English:

- **Call-outs.** Integers which refer to arrow labels in schematic diagrams should be tagged, so they will not be confused with numeric quantifiers.
- **Special forms.** Special phrases, such as chemical formulae, dates, addresses, and alphanumeric identifiers should be tagged and parsed with special grammar rules.
- **Measurement expressions.** Compound expressions of measure should be tagged to reduce parsing complexity, as in (10). “±” indicates the “ \pm ” symbol.

(10) <measure> <metric>42.931 ± 0.01 mm</metric>
<english>1.5902 ± .0005 inch</english> </measure>

Specific grammar rules which parse the open/close tags in nested constructions like this one guarantee that they will fire only in desired contexts, thereby limiting ambiguity.

5.8 On-line controlled authoring

In order to deploy controlled English for production authoring of technical text, an on-line system must be created for interactive checking of texts. This ensures that texts conform to the desired vocabulary and grammar constraints. An on-line authoring system can also support interactive disambiguation of lexical and structural ambiguities in the text. When problems are found, the author is asked to either rewrite parts of the sentence (with some help from the system) or answer questions about the sentence (to eliminate ambiguity). The result is a text which meets the constraints of controlled English, and encodes a single chosen meaning for each ambiguous lexical item or PP (prepositional phrase) attachment.

Once a controlled English vocabulary has been specified, it can be built into a vocabulary checking tool for on-line use by the author. For example, Caterpillar authors utilize an authoring workstation environment called *ClearCheck*, developed by Carnegie Group, which checks that the vocabulary in each sentence conforms to the controlled vocabulary. The vocabulary checker uses information about synonyms and ambiguous terms to notify the author when the use of a term may not be appropriate, and attempts to offer alternatives whenever possible. Documents do not conform to controlled English until they pass vocabulary checking.

The *ClearCheck* tool also provides a front-end for grammar checking. The controlled grammar is built into a grammar-checking component, provided by the analysis module of the *KANT* system. This grammar checker parses each sentence in the source text to determine if a valid analysis can be found. If no analysis can be produced, then the sentence does not conform to controlled English and the author is prompted to rewrite.

If more than one valid analysis is found for a sentence during grammar checking, the grammar checker will indicate whether a lexical ambiguity or a structural ambiguity is the cause. The *ClearCheck* tool then queries the author interactively, providing a choice of meanings for the word in question (lexical ambiguity) or the structure in question (PP-attachment ambiguity). *ClearCheck* then inserts an SGML tag into the sentence which captures this choice.

5.9 The translator's perspective

Professional translators who have experience translating controlled texts and post-editing machine translations of controlled texts agree that there are both advantages and disadvantages to using a CL. The most important advantages include the following:

- CL provides texts that are less ambiguous, and hence easier to translate. The simplified sentence structures encouraged by CL can generally be translated without re-reading the source text to clarify the meaning of an individual

sentence. The same is true of post-editing when the translator is correcting MT output.

- CL provides texts that are more accurate and consistent in their use of terminology. One of the biggest challenges faced by a translator when learning a new domain is the acquisition of a good glossary for the technical terms in that domain, and knowledge of what terms are appropriate in which contexts. A CL which provides explicit support for terminology lookup and terminology checking promotes more consistent terminology in the source text, which in turn makes the text easier to translate. Since machine translations of CL texts are also highly consistent in their translations of technical terminology, the use of a CL can be a big help to a translator who is still unfamiliar with the technical terminology in a domain.

The main challenges that translators face when using a CL include the following:

- CL can be repetitive. Although repetitive text is quite understandable, excessively repetitive text can be stylistically unacceptable from the translator's point of view. In such cases, translators tend to delete the repetitive text or use pronouns instead in the translation. Examples (11)–(14) illustrate the kinds of post-editing that Spanish translators sometimes feel is necessary for machine translation of controlled text. Each example shows the original source text (a), the raw MT output (b), and the post-edited output (c), with a comment as to the nature of the change made by the translator (d).

- (11) a. Raise the lift arms and lower the lift arms.
b. *Levante los brazos de levantamiento y baje los brazos de levantamiento.*
c. *Levante y baje los brazos de levantamiento.*
d. The translator deleted the first reference to *lift arms* (bold text).
- (12) a. The cylinder head assembly has one inlet valve and one exhaust valve for each cylinder.
b. *El conjunto de cabeza de cilindros tiene una válvula de admisión y una válvula de escape por cada cilindro.*
c. *El conjunto de culata tiene una válvula de admisión y una de escape por cada cilindro.*
d. The translator deleted the second reference to *valve* (bold text).
- (13) a. Clean the safety signs or replace the safety signs if you cannot read the words.
b. *Limpie los avisos de seguridad o reemplace los avisos de seguridad si no puede leer las palabras.*
c. *Limpie los avisos de seguridad o reempláce los si no puede leer las palabras.*

- d. The translator replaced the second reference to *the safety signs* with the pronoun *los* (bold text).
- (14) a. Install the level check plug and the filler plug.
b. *Instale el tapón de comprobación del nivel y el tapón de llenado.*
c. *Instale estos dos tapones.*
d. The translator replaced translation of *the level check plug and the filler plug* with *estos dos tapones* ('these two plugs').
- CL can be too uniform from a stylistic point of view. Sometimes CL enforces particular structures for certain phrases and sentences which do not translate well stylistically. In general, translators tend to vary the way they express the same kind of statement from paragraph to paragraph; they tend to be less consistent and more stylistically varied. Examples (15)–(17) illustrate the kinds of changes translators make to vary the style of CL translations. Again we show the original source text (a), the raw MT output (b), and the post-edited output (c).
- (15) a. As required, install the shims.
b. *Según se requiera, instale los calces.*
c. i. *Instale los calces que se requieran.*
ii. *Instale los calces necesarios.*
- (16) a. The electronic control has failed.
b. *El control electrónico ha fallado.*
c. i. *Falla del control electrónico.*
ii. *El control electrónico presenta fallas.*
- (17) a. For more information, consult your dealer.
b. *Para más información, consulte a su distribuidor.*
c. i. *Si desea más información, consulte a su distribuidor.*
ii. *Para obtener más información, consulte a su distribuidor.*

These issues are at the heart of the trade-off between productivity and stylistic excellence which translators will face when working with a CL. For technical text, it is generally acceptable to the end-user if a document is simple and repetitive, as long as its contents are accurate. Most end-users (for example, mechanics fixing heavy equipment) are not likely to read more than a few paragraphs of a manual at a time. Translators, however, tend to think of their work in holistic terms, and prefer to produce texts which flow from beginning to end with appropriate stylistic variation. Since a CL is typically introduced to improve consistency, reusability, and machine translatability of the source text, excessive post-editing by the translator can negate the advantages of the CL when working with technical text (such as

operation and maintenance manuals). For other types of text, where style is of utmost importance, the use of CL is less appropriate.

5.10 Design and deployment issues

When a CL is designed for an MT system, the constraints may be stricter than in a CL designed just for authoring. That is due to the fact CL for MT must do additional work to reduce ambiguity (and hence increase translation quality) as much as possible. As a result, we tend to focus on disambiguation of input sentences when developing a CL for MT. However, usability and author productivity are equally important. In this section, we discuss issues related to design and deployment of a CL.

5.10.1 Does controlling the source text really help?

When controlled English is introduced, the number of parses per sentence can be reduced dramatically. If a general lexicon and grammar are used to parse specialized domain texts, then analyses may be assigned which are not appropriate in the domain.

We have experimented with the *KANT* analyzer in order to determine the positive effects of the controlled English mentioned above (Baker et al., 1994). We used a test suite of about 750 sentences (part of a development/regression test suite for one *KANT* application). The sentences in the test suite range in length from 1 word to over 25 words. When a constrained lexicon and grammar for the domain were utilized, along with disambiguation by the author, the average number of syntactic analyses dropped from 27% to about 1%. About 95% of the sentences were assigned a single interlingua representation. Constraining the lexicon seems to achieve the largest reduction in the average number of parses per sentence. As expected, the best results are achieved when the system is run with constrained lexicon and grammar.

5.10.2 Expressiveness versus complexity

If we assume that the expressiveness of a language is some measure of the variety of lexical and grammatical constructions it allows, then the more expressive a language is the more complex it will be to analyze during translation. In some cases, however, reducing the expressiveness of a language does not necessarily reduce the complexity of analysis. In systems where the vocabulary is extremely limited (as, for example, in the earlier CFE), the authors may need to write long, convoluted sentence to express complicated meanings. In *KANT* Controlled English, the size of the vocabulary is not limited, and only those lexical or grammatical constructions which are unnecessarily complex are ruled out. The result is a language which is

expressive enough to author technical documents, but limited in complexity such that high-quality translations can be achieved.

5.10.3 Author involvement versus post-editing

An original goal in developing KANT Controlled English was to eliminate lexical ambiguity entirely. When this seemed impractical following domain analysis, it was decided to increase the amount of author involvement by introducing **interactive disambiguation**. Since the effect of ambiguity in the source text is reduced accuracy in the target text, increased post-editing is avoided when authors help to disambiguate the text. This is desirable in domains where the source language is translated to several target languages and increased cost of post-editing is prohibitive. In domains where there are fewer target languages, the other side of this trade-off might be explored, if the number of ambiguous terms and types of post-editing operations required allow cost-effective post-editing.

5.10.4 Controlled target-language definition

When a source document is authored in CL for MT, the translated document can be expected to have at best the same stylistic quality as the source document. However, this constraint is not always evident to customers, who often expect the output to be stylistically better than a sentence-for-sentence translation of the controlled source. Since CL promotes the writing of short, concise, sentences with redundancy (limited use of pronouns), the translated text will have similar style. To avoid unnecessary post-editing which aims at re-introducing a “non-controlled” style, it is important to have a CL specification for the target language, also. Creating such a specification, in direct correspondence with the controlled source-language definition, helps to set appropriate expectations about output quality.

5.10.5 CL maintenance

If we do not need to add, change or delete terminology once a CL is defined, then terminology maintenance is not a major issue. In a typical document-production operation, however, there is an ongoing need to update terminology due to the introduction of new products, new types of documents, etc. When a large number of authors (e.g. over 100) is simultaneously authoring documents using CL, it is important to have a well-defined language-maintenance process in place.

First of all, it is necessary to have a **problem-reporting** process that authors use when they encounter an apparent need for new terminology or grammar rules. When requests come directly from authors, it is essential to do initial terminology and grammar screening by an expert, since requests may come from a variety of authors with different levels of expertise. Sometimes, we find author requests to be

redundant or unnecessary. It is important to control the proliferation of terminology. If we do not implement a careful screening process, the terminology base will expand quickly to an unmanageable size. It is also important to have process monitoring and quality control through periodic review of source and target documents. Experienced editors who participate in a mentoring process for new authors can promote the integrity of CL standards.

Once the decision is made to update terminology, the CL checker should support rapid terminology update. The translation system must also support rapid update of the target-language terminology. Terminology update becomes a challenge if the amount of requests is large and the screening process becomes burdensome.

5.10.6 Success criteria for introducing CL

CL for MT works well when the following characteristics are present in the intended application domain:

- **Translation for dissemination.** When documents are authored in one language, in a particular domain, and are then translated into multiple languages, it is possible to control the style and content of the source text. This type of translation is referred to as “translation for dissemination”. A given domain is less amenable to a CL approach when unrestricted texts from multiple source languages are to be translated into one target language. This type of translation is referred to as “translation for assimilation”.
- **Highly-trained authors.** It may not be easy to deploy CL in an existing authoring process at first, because authors are used to writing texts in their own style for many years. Therefore, it is crucial for success that the authors are able to accept the notion of CL, and are willing to receive CL training. It seems that authors who receive comprehensive training and who use CL on a daily basis achieve the best results and highest productivity. It is also important that these well-trained authors act as mentors during the training of other authors new to CL. Adequate training and mentoring is crucial for author acceptance of CL.
- **Use of CL checkers.** Although CL can be implemented simply as a set of written guidelines for authors, uniform quality of CL text is maximized if the author uses a CL checker to write texts which are verified to comply with the CL definition. The use of an on-line checking system enhances consistency and promotes the reuse of texts across similar product lines where appropriate. Authored texts can also be aligned with their translations in a translation memory, leading to increases in production efficiency for technical authoring and translation.

- **Well-defined domain.** The success of a CL relies heavily on ruling out ambiguous meanings for terms which are not required in the given domain. Therefore, CL may be less suitable for unrestricted domains, such as general newsletters, email or bulletins. On the other hand, it is possible to control technical vocabulary and writing style in most technical documentation, since the domain is specific and it is preferable to standardize terminology and writing style.

6. Future directions

Our experience thus far has demonstrated that CL can have a significant positive impact on both authoring quality and translation productivity. Nevertheless, many challenges remain in an environment with a complex set of products and document types, and where terminology is updated constantly. The CTE application for Caterpillar has helped to advance the state of the art in CL systems, while simultaneously driving the research agenda for future work on new applications at CMU.

A number of issues in the field of CL deserve further attention. First and foremost, there is a clear need for more empirical evaluation. Present-day human-oriented CLs are often not specified very precisely and consistently. This causes confusion in their application and complicates evaluation. Furthermore, the motivation for individual writing rules is generally based on intuition rather than on empirical evidence. Thus, some writing rules may even do more harm than good. In fact, as we have seen, there is little empirical evidence to support the central claim that the use of CL does indeed improve readability and comprehensibility. Therefore, the area of CLs is in need of a more empirical foundation that requires a clear specification of the restrictions, so that application and evaluation can be more straightforward.

Perhaps an ideal situation for CL is for the machine to rewrite texts automatically into CL without changing the meaning expressed by the sentence. For example, vocabulary selection could be done automatically when the author uses a term outside the controlled vocabulary. Sentences would be rewritten if the author uses expressions outside the CL grammar. Furthermore, disambiguation would be done automatically with no author interaction. After the machine's rewrite is completed, the author would just read the text to confirm that it still expresses the original intention and that there are no major stylistic errors. Such a rewriting system could help to maximize author productivity and minimize training problems, while taking full advantage of the benefits of CL. In order to build such an automatic rewriting system, there are many research challenges which must be addressed.

There have been already some efforts towards automatic rewriting systems. For

example, in the LRE SECC project, a tool was designed which checks to see if documents comply with syntactic and lexical rules, and if not, then automatic correction is attempted wherever possible (see Adriaens, 1994). Another study proposes the use of a linguistic framework to produce paraphrases for certain constructions (Nasr et al., 1998). There has also been some research on automatic rewriting rules for Japanese–English MT (Shirai et al., 1993, 1998).

When we work towards an automatic rewriting system for CL, there are at least two different purposes. One is to assist the author in the publication of a CL text which is not translated. The other is for authoring input to an MT system. Both types of systems could be fully or partially automatic, depending on the requirements of the domain.

For a source-only rewriting system, phenomena such as disambiguation, pronoun reference and elliptical reference, which are difficult for MT, may not need to be resolved during the rewriting process. The focus is rather on grammatical, concise sentences, clarity of expression, and consistency of vocabulary usage, which help readers to understand the source document. A rewriting system may also be designed for non-native speakers of a language, who would like to check to see if their sentences comply with the grammar of the language.

An automatic rewriting system specifically for MT, on the other hand, can focus on internal rewriting rules, particular to source- and target-language characteristics, to make it easier to produce a high-quality MT. The input to MT would not necessarily be in human-readable form. Input sentence structure could be transformed to make it closer to target-language syntax when a system translates only to one target language. For example, automatic rewriting rules are often used for a Japanese–English MT system because the syntax of the two languages is very different and it is useful to transform the input sentences before running them through MT. An experiment in automatic rewriting shows that the quality of Japanese–English MT is improved by 20% when rewriting rules are applied (Shirai et al., 1998).

Since the *KANT* system is designed to support both of these purposes, an automatic rewriting system must produce both publication-quality text and fully disambiguated input sentences for multilingual MT. For example, input sentences for MT may use redundant references, such as full noun phrases instead of pronouns, where publication quality text might use a pronoun instead of repeating a noun phrase.

A CL for MT attempts to rule out difficult sentence structures and to limit ambiguous vocabulary items in order to achieve accurate translation. However, if a CL becomes too restrictive, it may introduce usability and productivity problems. If it is too difficult to write sentences that comply with the CL, no one will use it. Controlled sentences which are not stylistically adequate will not be accepted by

authors and will be heavily post-edited by translators. Therefore, it is essential to find a middle ground which is productive and acceptable for authors and which promotes high-quality translation. In order to improve author productivity, it is desirable to develop an automatic rewriting system to convert text into CL. For the field of CL, this will be a new challenge and a future direction of research and development.

Further reading

The material in Section 1 is adapted from Huijsen (1998). The material in Section 1.4 is partly based on Adriaens and Schreurs (1992). Much of the material in Section 5 is adapted from previously published papers, notably Kamprath et al. (1998), Mitamura and Nyberg (1995), and Mitamura (1999).

An obvious source of material is the proceedings of the Controlled Language Applications Workshop, of which there have so far been two, in Leuven, Belgium (CLAW, 1996) and in Pittsburgh (CLAW, 1998).

References for individual CLs are to be found in CLAW (1996, 1998), and additionally the following: Douglas Aircraft Company (Gringas, 1987); Airspeak (Robertson and Johnson, 1987); Seaspeak (Glover et al., 1983). For CTE see Kamprath et al. (1998). The “SE Guide” is AECMA (1995). For a detailed discussion of the SE lexicon, see Humphreys (1992). A thorough critique of SE is given by Lehrndorfer (1992).

References for CL checkers are in CLAW (1996, 1998), and additionally as follows: Adriaens and Macken (1995) report on the SECC project. The development of the BSEC is described by Wojcik et al. (1990, 1993) and Hoard et al. (1992).

The use of a CL in combination with MT by Perkins Engines Ltd. is described by Pym (1990) and Newton (1992). See van der Eijk et al. (1996) and de Koning (1996) for a discussion of the work at Cap Gemini’s Lingware Services.

Notes

* The authors would like to express their gratitude to Enrique Torrejón for the examples and commentary on controlled language from the translator’s perspective.

1. “Conversion” is the process which derives words without changing the phonological form. For example, the verb *pump* is derived from the phonologically identical noun *pump*.

2. Examples taken from the AECMA Simplified English lexicon (AECMA, 1995).

3. Mitamura and Nyberg (1995), Nyberg and Mitamura (1996), Lux and Dauphin (1996).

4. Bibliographic references for CL systems are given in the “Further reading” section.
5. Carnegie Group was acquired by Logica in 1997.
6. Mitamura et al. (1991), Carbonell et al. (1992), Nyberg and Mitamura (1992).
7. Of course, one could claim that pattern substitution is a primitive way to do MT.
8. Bibliographic references for CL checkers are given in the “Further reading” section.
9. See de Koning (1996), van der Eijk et al. (1996).
10. See Wojcik et al. (1990) and Adriaens and Macken (1995). Fouvry and Balkan (1996) discuss the creation of test suites for this type of evaluation.

References

- Adriaens, Geert (1994) “Simplified English Grammar and Style Correction in an MT Framework: The LRE SECC Project”, in *Translating and the Computer 16*, London: Aslib, pages 78–88.
- Adriaens, Geert and Lieve Macken (1995) “Technological Evaluation of a Controlled Language Application: Precision, Recall and Convergence Tests for SECC”, in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation: TMI 95*, Leuven, Belgium, pages 123–141.
- Adriaens, Geert and Dirk Schreurs (1992). “From COGRAM to ALCOGRAM: Toward a Controlled English Grammar Checker”, in *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, COLING-92*, Nantes, France, pages 595–601.
- AECMA (1995) *A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language — Issue 1* (AECMA document PSC-85-16598, commonly known as “The Simplified-English Guide”), Brussels, Belgium.
- Baker, Kathryn L., Alexander M. Franz, Pamela W. Jordan, Teruko Mitamura, and Eric H. Nyberg 3rd (1994) “Coping with Ambiguity in a Large-Scale Machine Translation System”, in *COLING 94: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pages 90–94.
- Carbonell, Jaime G., Teruko Mitamura, and Eric H. Nyberg 3rd (1992) “The KANT Perspective: A Critique of Pure Transfer (and Pure Interlingua, Pure Statistics,...)”, in *Quatrième Colloque international sur les aspects théoriques et méthodologiques de la traduction automatique, Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, TMI-92*, Montréal, Canada, pages 225–235.
- Chervak, Steven, Colin Drury and J. P. Ouellette (1996) “Field Evaluation of Simplified English for Aircraft Workcards”, in *Proceedings of the 10th Federal Aviation Administration / Association for Aerospace Medicine Meeting on Human Factors in Aviation Maintenance and Inspection*, Alexandria, VA.
- CLAW (1996) *Proceedings of the First International Workshop on Controlled Language Applications CLAW 96*, Leuven, Belgium.
- CLAW (1998) *Proceedings of the Second International Workshop on Controlled Language Applications CLAW 98*, Pittsburgh, Pennsylvania.

- de Koning, Michiel (1996) "Bringing Controlled Language Support to the Desktop", in *Proceedings of the 1996 European Association for Machine Translation (EAMT) Conference*, Vienna, Austria, pages 11–19.
- Fouvry, Frederik and Lorna Balkan (1996) "Test Suites for Controlled Language Checkers" in CLAW (1996), pages 179–192.
- Fuchs, Norbert E. and Rolf Schwitter (1996) "Attempto Controlled English (ACE)" in CLAW (1996), pages 124–136.
- Glover, Alan, Edward Johnson, Peter Strevens and Fred Weeks (1983) *Seaspeak Training Manual*. New York: Pergamon Press.
- Goyvaerts, Patrick (1996) "Controlled English, Curse or Blessing? A User's Perspective", in CLAW (1996), pages 137–142.
- Gringas, Becky (1987). "Simplified English in Maintenance Manuals", *Technical Communication* 34, 24–28.
- Hoard, James, Richard Wojcik, and Katherine Holzhauser (1992) "An Automated Grammar and Style Checker for Writers of Simplified English", in Patrik Holt and Noel Williams (eds), *Computers and Writing: State of the Art*, Dordrecht, The Netherlands: Kluwer Academic Publishers, pages 278–296.
- Holmback, Heather, Serena Shubert, and Jan Spyridakis (1996) "Issues in Conducting Empirical Evaluations of Controlled Languages", in CLAW (1996), pages 166–177.
- Huijsen, Willem-Olaf (1998) "Controlled Language – An Introduction", in CLAW (1998), pages 1–15.
- Humphreys, Lee (1992) "The Simplified English Lexicon", in *Proceedings of the European Association for Lexicography Congress (EURALEX)*, Tampere, Finland, pages 353–364.
- Kamprath, Christine, Eric Adolphson, Teruko Mitamura and Eric Nyberg (1998) "Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English", in CLAW (1998), pages 51–61.
- Kincaid, Calliopi (1997) *A Validation Study of Simplified English as a Facilitator in English for Special Purpose Language Learning*. Technical report, University of Central Florida, Orlando, Florida.
- Lehrndorfer, Anne (1992) *Simplified English von AECMA – Beschreibung und Diskussion*, Technical Report CIS-Bericht-92–60, Centrum für Informations- und Sprachverarbeitung, Universität München, Germany.
- Lux, Veronika and Éva Dauphin (1996) "Corpus Studies: a Contribution to the Definition of Controlled Language", in CLAW (1996) pages 193–204.
- Mitamura, Teruko (1999) "Controlled Language for Multilingual Machine Translation", in *Machine Translation Summit VII '99*, Singapore, pages 46–52.
- Mitamura, Teruko and Eric H. Nyberg 3rd (1995) "Controlled English for Knowledge-Based MT: Experience with the KANT System", in *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation: TMI 95*, Leuven, Belgium, pages 158–172.
- Mitamura, Teruko, Eric H. Nyberg 3rd, and Jaime G. Carbonell (1991) "An Efficient Interlingua Translation System for Multi-lingual Document Production", in *Machine Translation Summit III, Proceedings*, Washington, DC, pages 55–61.

- Nasr, Alexis, Owen Rambow, and Richard Kittredge (1998) “A Linguistic Framework for Controlled Language Systems”, in CLAW (1998), pages 145–158.
- Newton, John (1992) “The Perkins Experience”, in John Newton (ed.), *Computers in Translation: A Practical Appraisal*, London: Routledge, pages 46–57.
- Nyberg, Eric H. III and Teruko Mitamura (1992) “The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains”, in *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, COLING-92*, Nantes, France, pages 1069–1073.
- Nyberg, Eric H. 3rd and Teruko Mitamura (1996) “Controlled Language and Knowledge-Based Machine Translation: Principles and Practice”, in CLAW (1996), pages 74–83.
- Ogden, Charles (1932). *Basic English, A General Introduction with Rules and Grammar*. London: Paul Treber and Co.
- Ogden, Charles (1942) *The General Basic English Dictionary*. New York, NY: W. W. Norton and Co.
- Pym, P. J. (1990) “Pre-editing and the Use of Simplified Writing for MT: An Engineer’s Experience of Operating an MT System”, in Pamela Mayorcas (ed.), *Translating and the Computer 10: The Translation Environment 10 Years on*, London: Aslib, pages 80–95.
- Robertson, Fiona and Edward Johnson (1987) *Air Traffic Control Language — Pilot Training*. Hillsdale New Jersey: Prentice-Hall.
- Shirai, Satoshi, Satoru Ikehara and Tsukasa Kawaoka (1993) “Effects of Automatic Rewriting of Source Language within a Japanese to English MT System”, in *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation: TMI ’93*, Kyoto, Japan, pages 226–239.
- Shirai, Satoshi, Satoru Ikehara, Akio Yokoo, and Yoshifumi Ooyama (1998) “Automatic Rewriting Method for Internal Expressions in Japanese to English MT and Its Effects”, in CLAW (1998), pages 62–75.
- Shubert, Serena, Jan Spyridakis, Heather Holmback and Mary Coney (1995) “The Comprehensibility of Simplified English in Procedures”, *Journal of Technical Writing and Communication* 25, 347–369.
- van der Eijk, Pim, Michiel de Koning, and Gert van der Steen (1996) “Controlled Language Correction and Translation”, in CLAW (1996), pages 64–73.
- Wojcik, Richard, James Hoard and Katherine Holzhauser (1990) “The Boeing Simplified English Checker”, in *Proceedings of the International Conference on Human-Machine Interaction and Artificial Intelligence in Aeronautics and Space*, Centre d’Etudes et de Recherche de Toulouse, Toulouse, France, pages 43–57.
- Wojcik, Richard H., Philip Harrison, and John Bremer (1993) “Using Bracketed Parses to Evaluate a Grammar Checking Application”, in *31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, pages 38–45.

CHAPTER 15

Sublanguage

Harold Somers
UMIST, Manchester, England

1. Introduction

In the previous chapter, the authors discussed the idea that by imposing controls on the input to MT, better quality output could be achieved. In this chapter, we discuss a superficially similar idea in which, again, a reduced lexicon and restricted set of syntactic structures means that we can achieve a much higher quality of MT output. The crucial difference between the controlled-language approach of the previous chapter, and the “sublanguage” approach to be discussed in this chapter is that, whereas the restrictions of controlled language are imposed on the authors, those of a sublanguage occur naturally.

It has long been recognised that the subject matter of a text affects not only the choice of vocabulary, but also the “style” of expression. This effect has been termed “register” by sociolinguists, while other terms used to denote a similar phenomenon include “special language”, “language for special purposes” (LSP) and, sometimes with pejorative overtones, “jargon”. The term **sublanguage**, usually used in connection with MT, dates back to Zellig Harris, the structuralist linguist, who gave a precise characterization of the idea in terms of his linguistic theory.¹ The term was coined with the mathematical idea of “subsystem” in mind, the “sub-” prefix indicating not inferiority, but inclusion. So a sublanguage is a subset of the “whole” language.

Like controlled language, a sublanguage approach to MT (and many other computational linguistics tasks) benefits from the two main characteristics of sublanguage as compared to the whole language, namely the reduced requirement of coverage in the lexicon and grammar. We will look into this in a little more detail in the next section. In computational linguistics, the sublanguage approach was pioneered by researchers at New York University in the later 1960s, lead by Naomi Sager. In MT, the idea was taken up in the 1970s at the Université de Montréal with the development of the *Météo²* system to translate weather bulletins from English into French, which has become the classical case of MT that works. We will look at

how *Météo*'s success is achieved as a case study below. Since the success of *Météo*, developers have been on the look-out for other sublanguages that would be suitable for MT. Some commentators have suggested that such cases are few and far between, while others argue that there are many such applications, and the number is growing as the World Wide Web-based information explosion encompasses users who wish to browse and surf in their own language. Among the applications that have seen some success we can cite job ads, military telexes, recipes, stock-market reports, abstracts of technical reports, avalanche warning bulletins, medical reports, messages between security and law enforcement agencies, and so on.

2. Properties of sublanguage

Let us begin with a definition of “sublanguage”:

The term *sublanguage* has come to be used ... for those sets of sentences whose lexical and grammatical restrictions reflect the restricted sets of objects and relations found in a given domain of discourse. (Kittridge and Lehrberger, 1982: 2)

This definition emphasizes the link between a sublanguage and the “domain of discourse” in which it is used. In other words, a sublanguage arises when a community of users — domain specialists — communicate amongst themselves. They develop their own vocabulary, that is not only specialist terms which have no meaning to outsiders, but also (and crucially) everyday words are given narrower interpretations, corresponding to the concepts that characterize and define the domain. In addition, there will be a favoured “style” of writing or speaking, with preferred grammatical usages.

2.1 The lexicon

It is hardly necessary to illustrate the idea that sublanguage vocabulary is specialized: translators are all too familiar with the problem of technical terms which are simply not found in general dictionaries, e.g. *erythrophleum*, *dyspnea*, *ptosys*, *organomegaly*, *rhonchi*, which are all medical terms. Everyday words too may take on a special meaning, e.g. in computer science *bit*, *browse*, *bus*, *log*, *mouse*, and so on. Occasionally, words commandeered in this way undergo grammatical changes. Thinking again of computer terminology, we have seen *mouses* as the plural of *mouse*, *inputted* as the past tense of *input*, and *Windows* as a singular noun (as in *Windows is...*). We will see in Section 2.2 that words “behave” differently in sublanguages too.

There is an obvious terminological aspect to sublanguages, and, as we discussed in Chapter 4, an important feature of terminology is that meanings (and translations) are often fixed, and neologism is closely controlled. However, not all the vocabulary in a technical text will necessarily have the status of terms, yet still there is evidence that the range of everyday vocabulary found within a given sublanguage is highly characteristic, and the usage can be specialised. For example, stock-market reports typically use a wide range of synonyms to express upwards and downwards movement, and can be classified according to whether they indicate large or small movements, or are neutral, as Figure 1 illustrates.

	neutral	large	small
upwards	<i>move up, advance, gain, rise, climb, push upward</i>	<i>jump, soar, surge, bounce up, spurt, shoot up</i>	<i>edge up, creep up, firm, struggle upwards</i>
downwards	<i>move down, fall, dip, drop, decline, slide</i>	<i>plunge, tumble, nosedive</i>	<i>drift down, slip, sag, ease, settle down</i>

Figure 1. Examples of movement words in stock-market reports
(from Kittredge, 1982: 118)

One claim that is sometimes made is that the vocabulary of a sublanguage tends towards finiteness, that is, given sufficient text of the appropriate type, we can extract all the vocabulary we are likely to meet in the future. In fact this is only partly true, and depends greatly on the subject field. For a start, almost any text may contain new “unknown” words in the form of proper names. This applies even in a relatively restricted sublanguage like recipes, where there is limitless scope for naming new dishes. On the other hand, notice how the list of possible ingredients, possible utensils, and possible things to do with them *is* relatively restricted. The size of the lexicons for sublanguages can of course vary hugely. The weather-bulletin sublanguage which we will discuss below is reportedly based on a lexicon of less than 1,000 words, not including place names. A set of aircraft maintenance manuals contained 40,000 different words.

The other major advantage with a sublanguage approach to the lexicon is the reduction of possible homonyms and ambiguities. Both in their use of technical terms and everyday words, sublanguages can be characterised as allocating favoured meanings to otherwise ambiguous words. This is the case, as already illustrated, with everyday terms that have a special meaning.³ But it is also the case that everyday words, although not technical terms, may be used with a preferred sense. For example in medical reports, *complain of* is used as a synonym of *report*, as in (1a), rather than with any notion of ‘make an accusation’ as in (1b).

- (1) a. The patient complained of stomach ache.
b. The patient complained of having to wait for an appointment.

Also, words which are grammatically ambiguous in that they can belong to different syntactic categories often appear predominantly in only one usage. For example, in recipes, *cook* is generally a verb rather than a noun, even though the noun meaning is quite appropriate for the domain. The potential ambiguity in (2) disappears if we suppose it refers to a piece of equipment that has a cover as one of its parts, and for which the action of covering something is never required: the noun reading of *cover* is preferred over the verb interpretation.

- (2) Remove bulb and cover.

2.2 Syntax

So far we have concentrated on the lexical aspect of sublanguages, but sublanguages can also be characterised by the syntactic constructions that they use, in parallel with the lexical features: a reduced range of constructions, some more favoured than others, preferred interpretations for ambiguous constructions, and, of special interest, “deviance” from standard constructions.

The reduction in the range of constructions used reflects the specific text-type and discourse of the sublanguage. As an example, it was found that in an aircraft maintenance manual there were no direct questions (3a), tag questions (3b), no use of the simple past tense (3c), nor any exclamatory sentences (3d).⁴

- (3) a. Do you have your tool kit? Is the motor turned off?
b. Check the batteries, won’t you? The switch should not be on, should it?
c. The engine stopped. High temperatures caused buckling.
d. How powerful the engine is! What a complex hydraulic system this plane has!

On the other hand, some constructions are particularly prevalent. An obvious example is a recipe, one part of which is largely made up of imperative sentences (4). Weather bulletins have a preference for future tenses (5).

- (4) a. Peel and chop the onions.
b. Fry gently with the butter in a deep frying-pan.
c. Sprinkle with flour and leave to simmer.
- (5) a. Tomorrow will be sunny.
b. Scattered showers will turn heavy later.
c. Temperatures will fall sharply overnight.

Because certain constructions are (dis-)favoured, this means that ambiguous constructions can be more confidently interpreted. If (6) is a job ad, it will be interpreted as a passive construction with a missing auxiliary verb although the active interpretation is a simpler construction.

- (6) Chef wanted to work in busy restaurant.

A particular feature of some sublanguages is that they permit “deviant” constructions, that is constructions which under normal circumstances would seem odd. For example, in medical reports, the verb *present* can occur without a direct object, with the meaning ‘appear in the surgery’ (7a). Airline pilots tend to use the word *overhead* as a preposition as in (7b). On British trains you can now hear announcements like (7c) in which the word *forward* apparently indicates the end-station. In weather forecasts, sentences often lack a main verb (7d).

- (7) a. The patient presented with a sore throat.
b. We are now flying overhead Paris.
c. This is the 9.15 train going forward to London Euston.
d. Cloudy tomorrow becoming rain.

Other characteristic features of the sublanguage stem from the particular meaning that individual words can have. For example, gold cannot normally plunge, nor can oil climb, unless it is in a stock-market report.

2.3 Text-type

Sublanguages are characterised by lexical and syntactic features, as we have seen; but a third, textual, parameter plays an important part. In fact, sublanguages are usually described by the subject domain, which determines the vocabulary and, to a certain extent, the syntax, and the **text-type**, which will account for other aspects of the syntax, and features of document structure. Text-type is determined by the medium (spoken or written), the author and reader, and other features of the communication process. We can easily distinguish gross text-types such as reports, manuals, lists of instructions, descriptive text, bulletins, scripts (texts meant to be spoken), transcripts (speech recorded as text), and so on. Each of these text-types has its own distinctive features, and in connection with a particular subject domain will determine the nature of the sublanguage. We can thus distinguish different sublanguages which are related by domain, e.g. spoken and written weather bulletins, more or less verbose versions of the same instruction manual, reports of the same incident from different perspectives, and so on. We can also see commonalities due to text-type across different domains. So for example, a telegraphic style in which definite articles, copula verbs and pronouns are omitted is common to

instruction lists, no matter what the subject domain, as the examples in (8) show.

- (8) a. Check indicator rod extension.
b. Leave motorway at next exit.
c. Peel and slice onions.
d. Check reservoir full.
e. Avoid south-facing slopes.
f. If too thick, add water.
g. Separate egg-whites into bowl and beat until stiff.
h. Remove gasket and clean.

2.4 Sublanguages and MT

Just as with controlled languages, many of the features of sublanguages can prove advantageous in developing MT or CAT systems. Unlike controlled languages however, where the restrictions may be motivated by the limitations of the MT system, with sublanguages it is the other way round: the restrictions occur naturally, and the MT system design can take advantage of them.

Not all sublanguages are necessarily good for MT. Some, for example, have features which actually make MT *more* difficult. For example, the aviation maintenance manual sublanguage investigated by the Montreal team was found to contain long noun sequences (9) which were very difficult to analyse automatically.⁵

- (9) a. external hydraulic power ground test quick-disconnect fittings
b. fuselage aft section flight control and utility hydraulic system filter elements
c. fan nozzle discharge static pressure water manometer

An extreme example of a distinctive sublanguage which has numerous features which are quite MT-*unfriendly* is “legalese”.⁶ So as to avoid the dangers of legal loopholes, legalese often involves cases of extreme precision (10). Another characteristic is a preference for nominalizations, omitted relative markers, truncated passives and multiple embeddings (11).

- (10) Know ye that I, ___, of ___, for and in consideration of ___ dollars, to me in hand paid by ___, do by these presents for myself, my heirs, executors, and administrators, remise, release and forever discharge ___, of ___, his heirs, executors, and administrators, of and from any and all manner of action or actions, cause and causes of action, suits, debts, dues, sums of money, accounts, reckonings, bonds, bills, specialties, covenants, contracts, controversies, agreements, promises, trespasses, damages, judgments, executions, claims, and demands whatsoever....

-
- (11) ...and to consent to immediate execution upon any such judgment and that any execution that may be issued on any such judgment be immediately levied upon and satisfied out of any personal property of the undersigned ... and to waive all right of the undersigned ... to have personal property last taken and levied upon to satisfy any such execution.

Clearly then, in searching for a sublanguage application suitable for MT, we can look for both pros and cons before deciding to develop a system. One of the advantages that has been claimed for the sublanguage approach is that cross-lingually there are similarities that can be exploited (see Kittredge, 1982). Since sublanguages express the concepts and relationships of the subject domain that they describe, it might not be surprising to find parallels between equivalent sublanguages in two different languages. While the particular structures might not be exactly equivalent (for example, recipes use an imperative form in English but an infinitive in French), there are broad similarities. When some construction is absent in one sublanguage, its equivalent is generally absent from the other-language counterpart. And where parallel structures can be identified, their relative frequency has been found to be similar.

In the next section of this chapter, we will look at perhaps the classic case of sublanguage MT, the *Météo* system.

3. *Météo*: a case study

Research on NLP began at the Université de Montréal in 1965 under the direction of Guy Rondeau. At this time, the Canadian government introduced its bilingual policy, requiring all official documentation to be available in both English and French. The demands on the translation service grew considerably, and the Canadian National Research Council agreed to look into the possibilities of MT. Between 1968 and 1971 the research group TAUM (*Traduction Automatique de l'Université de Montréal*) developed a prototype English–French system, and in 1975 received a contract to tackle the translation of weather bulletins. A first version of the system was demonstrated in 1976, and *Météo* began full-time operation in May 1977. Since that date, the weather bulletins transmitted daily by Environment Canada have been largely translated by MT. In October 1984 the system, now the property of John Chandioux Consultants Inc., was reinstalled on four microcomputers, and since then has been extended to cover agricultural and maritime bulletins as well as regular weather reports. *Météo* has translated more than 30 million words, with less than 5% requiring any human intervention whatsoever. As well as being very inexpensive (0.5 cents per word without postediting) and, of

course, quick (an average bulletin is 250 words long, but takes only four minutes to be translated automatically), without the MT solution the translations would simply not be done at all.⁷

The task is in many respects ideal for automation. Weather forecasts are produced at numerous sites across the nation at regular intervals throughout the day, and are thus relatively short-lived. Translating them is both very boring and highly repetitive, with low job satisfaction and a resultingly high turnover of staff employed to do the job.

3.1 How *Météo* works

The (sub-)language of weather bulletins is indeed highly restricted. To the casual observer it might be believed that the way to translate these bulletins automatically is to have a list of all the turns of phrase that occur, list them together with their translations, and simply string them together as needed. In fact, this is not practical, even with as restricted a sublanguage as that of weather bulletins. Although there are only a small number of phrases, they can nevertheless be combined in an almost limitless number of ways. Taking into account also the fact that they may contain place names, and numerical data such as temperatures or wind-speeds, it becomes obvious that the simple cut-and-paste approach will not work.

Instead, *Météo* employs the methods of the so-called “second generation” of MT systems, subjecting the input text to a grammatical analysis and then adapting (“transferring”) the resulting representation into a form appropriate for generating the French. In this description, we will skip some of the less interesting details, but aim to give an overall impression of how it works.

A typical weather report is received in a standard format, as illustrated in Figure 2: a coded heading, a statement of the origin of the bulletin, a list of regions to which the bulletin applies, the forecast itself, and then a terminator to indicate the end of the bulletin.

```
FPCN11 CYYZ 311630
FORECASTS FOR ONTARIO ISSUED BY ENVIRONMENT CANADA AT 11.30
AM EST WEDNESDAY MARCH 31ST 1976 FOR TODAY AND THURSDAY.
METRO TORONTO
WINDSOR.
CLOUDY WITH A CHANCE OF SHOWERS TODAY AND THURSDAY.
LOW TONIGHT 4. HIGH THURSDAY 10.
OUTLOOK FOR FRIDAY ... SUNNY
END
```

Figure 2. Weather report as received

An initial pass will separate the bulletin into units, at the same time searching for “unknown words” — normally the result of mistyping at source — which would mean that the text would have to be translated by hand.

The next stage involves dictionary look-up which identifies linguistic categories (such as noun, verb and so on) and associated grammatical and semantic features. These features are used in the “parsing” process, and are tailored to the sublanguage so as to include, as well as the expected things like number (singular, plural), features that are specifically relevant. For example, the word *heavy* has different translations in French depending on whether the thing so described is a stationary meteorological phenomenon (e.g. fog, clouds, in which case *dense*), falling (rain, snow, hail: *abondant*) or blowing (wind, gale: *fort*). Adjectives and nouns, but also prepositions and other function words are marked for these features which include distinctions between time-point, time-duration, place, direction, measure and so on.

The parsing stage recognises just five different possible syntactic structures. The first, which is essentially just a template with gaps for place-names, dates and times covers the “header” information as in (12).

- (12) FORECASTS FOR <place> ISSUED BY ENVIRONMENT CANADA AT
<time> <timezone> <day> <date> FOR <time-period>.

The next is simply a list of place names. The third, and most flexible structure, is a meteorological condition. The condition itself is expressed by a noun or adjective such as *rain*, *cloudy* — the grammatical distinction is not important for this system — and may be modified by phrases before and/or after the head-word (13), restricted to a certain location or time period, or both (14), and any of these can be coordinated with *and* or *or* (15).

- (13) a. heavy rain
- b. sunny with moderate winds
- c. mostly cloudy with a chance of showers
- (14) a. snow with flurries in coastal regions
- b. heavy winds this evening
- c. mainly sunny with moderate winds in exposed regions later
- (15) a. heavy rain or snow on higher ground
- b. bright periods today and tomorrow

The fourth structure is similar to the third, but preceded by a phrase such as *outlook for tomorrow*. The same range of statements is possible here, but additionally we get phrases like *continuing*, *turning*, *becoming*. The final basic structure type expresses value ranges such as temperatures (highs and lows). Again these can be modified by

time and location limiters, and can be coordinated.

The parsing phase uses a computational rule formalism to analyse the input text into one of these structures. In doing so, it effectively disambiguates most of the few ambiguities that the sublanguage allows. For example, *heavy* is an ambiguous word in this sublanguage, given its three alternative translations in French. Similarly, *morning* can be *matin* if it is a time-point, but *matinée* if a time duration. The preposition *around* is translated as *environ* for temperatures but *vers* for times. Some structures are also ambiguous. For example in (15a), does the location *on higher ground* refer only to snow or to the unsure *rain or snow* condition? Sometimes, French allows much the same imprecision, so the ambiguity can pass unresolved, though compare (16a) where there is no ambiguity (because snow does not gust), and (16b) where the two interpretations will have different translations (does the adjective apply to both nouns?).

For the generation of the French text, some of the structures need to be transformed. Some weather conditions described by an adjective in English are expressed as a noun in French, e.g. *cloudy periods* might be translated as *parfois nuageux* ('sometimes cloudy'). In general, some modifiers which precede the term in English must be transposed which sometimes introduces complications: compare (17a,b).

- (17) a. heavy rain from the south
 → *pluie abandonante du sud*
 rain heavy from-the south
 b. strong gusty southerly winds
 → *vents forts du sud soufflant en rafales*
 winds strong from-the south blowing in gusts

Otherwise, the generation of the French text involves agreement of adjectives and nouns, choice of preposition (cf. *à Montréal* but *en Nouvelle-Écosse, au Manitoba*), and elision and contraction (e.g. *à + le* → *au*).

The computational details of how *Météo* works are probably of limited interest to readers of this book. Briefly, the linguistic facts are expressed as “rules” in a kind of formalism which is supposed to be relatively easy for linguists to learn and understand. There is a computer program which takes these rules and applies them to the input text, building up the structure. The computer program is independent of the particular set of rules that have been written, which means that it has always been relatively straightforward to amend and update the program over the years.

3.2 Twenty-five years on

Météo was remarkable for many reasons. One of them was the speed with which the developers went from the basic blueprint to a working system. Once the system was working, the “concept” was established, and they were able to develop other versions of essentially the same system. Different types of weather forecasts (maritime gale warnings, weather for farmers) and translations between different language pairs (French–English, notably, but there has also been talk of developing the system for Inuktitut) might be more or less easy to develop on the basis of *Météo*.

One such development was for the translation of weather synopses. While both types of texts (bulletins and synopses) ostensibly deal with the same subject matter, it was found that the range of vocabulary is much greater in synopses (between 2,000 and 4,000 words, compared to 1,000 for the bulletins). Furthermore, while the purpose of the bulletins is to report the latest forecast as concisely as possible, synopses give a more general summary with less emphasis on brevity, as example (18)⁸ illustrates.

- (18) Variable skies and isolated showers were reported overnight and this pattern is forecast to continue through this morning across Southern Ontario. The weak disturbance responsible for this weather will move east of the region by tonight allowing skies to clear once again. High pressure will dominate the weather picture on Thursday bringing sunny skies, light winds and temperatures several degrees above seasonal norms.

Researchers in Montreal were also interested in the possibility of automatically generating (multilingual) weather reports directly from the raw meteorological data, though this is clearly not translation as such, and involves numerous problems of little interest to us.

A perhaps more interesting development was reported in connection with a contract that was awarded at the time of the Atlanta Olympic Games in 1996 to provide French translations of weather bulletins to athletes and visitors. The basic design of the *Météo* system meant that it was feasible to transport the system from Canada to the southern USA and spend only two weeks customizing it. The details of this “customization” make interesting (and amusing) reading. First, perhaps obviously, there are a number of weather conditions that apply to Atlanta that do not occur in Canada (and, *vice versa*, a number of words which could be taken out of the dictionary, especially considering that the system would be used only for two weeks in the summer). Along the same lines, the sublanguage of weather bulletins has some differences between American and Canadian English. John Chandioux tells the amusing story of how one of the meteorologists, who insisted on spelling *tommorrow* thus (with two *ms*) despite being corrected on three consecutive days,

was accommodated by having *tommorrow* added to the lexicon. Some other turns of phrase differed too. Also of interest was the fact that over the years since *Météo* was first developed, new aspects of the weather had come to be included in the reports: UV index, pollen count, wind chill and so on. Each of these developments required additions — albeit simple to program — to the system.

Further reading

The most concentrated source of material on sublanguages can be found in two edited collections: Kittredge and Lehrberger (1982) and Grishman and Kittredge (1986). Many of the examples given in this chapter are taken from those sources.

Of relevance is Sager et al.'s (1980) discussion of "special languages". Sager (1993) has some interesting views on the notion of "text-type", especially with reference to translation. Biber (1988, 1995) discusses "register" from a computational point of view.

Detailed descriptions of *Météo* can be found in Hutchins (1986), pages 228–231, Isabelle (1987), Chapter 12 of Hutchins and Somers (1992), or Chapter 7 of Whitelock and Kilby (1995).

Notes

1. Harris (1968). An early use of the term is also found in Sager (1975).
2. *Météo* is a registered trade mark.
3. Although we cannot exclude the possibility of interference. For example, one could imagine the user manual for a computer game which involved a mouse as its central character....
4. All examples are from Lehrberger (1982: 84).
5. These examples are from Lehrberger (1982: 92).
6. Examples of legal sublanguage are from Charrow et al. (1982).
7. The figures in this paragraph are quoted from a 1997 article by John Chandioux, president of John Chandioux Consultants Inc., from *Observatoire Québécois des Industries de la Langue*, <http://199.84.130.134/oql/Tao/tradauto.htm>.
8. This example is from Lehrberger (1986: 28).

References

- Biber, Douglas (1988) *Variation across Speech and Writing*. Cambridge: Cambridge University Press.
- Biber, Douglas (1995) *Dimensions of Register Variation: A Cross-Linguistic Comparison*. Cambridge: Cambridge University Press.
- Charrow, Veda R., Jo Ann Crandall and Robert P. Charrow (1982) "Characteristics and Functions of Legal Language", in Kittredge and Lehrberger (1982), pages 175–190.
- Grishman, Ralph and Richard Kittredge (eds) (1986) *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Harris, Z. (1968) *Mathematical Structures of Language*. New York: Wiley.
- Hutchins, W. J. (1986) *Machine Translation: Past, Present, Future*. Chichester, Sussex: Ellis Horwood.
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*. London: Academic Press.
- Isabelle, Pierre (1987) "Machine Translation at the TAUM Group", in Margaret King (ed.), *Machine Translation Today: The State of the Art*, Edinburgh: Edinburgh University Press, pages 247–277.
- Kittredge, Richard (1982) "Variation and Homogeneity in Sublanguages", in Kittredge and Lehrberger (1982), pages 107–137.
- Kittredge, Richard and John Lehrberger (eds) (1982) *Sublanguage: Studies of Language in Restricted Semantic Domains*. Berlin: Walter de Gruyter.
- Lehrberger, John (1982) "Automatic Translation and the Concept of Sublanguage", in Kittredge and Lehrberger (1982), pages 81–106.
- Lehrberger, John (1986) "Sublanguage Analysis", in Grishman and Kittredge (1986), pages 19–38.
- Sager, Juan C. (1993) *Language Engineering and Translation: Consequences of Automation*. Amsterdam: John Benjamins.
- Sager, Juan C., David Dungworth and Peter F. McDonald (1980) *English Special Languages: Principles and Practice in Science and Technology*. Wiesbaden: Brandstetter.
- Sager, Naomi (1975) "The Sublanguage Technique in Science Information Processing", *Journal of the American Society for Information Science* 26, 10–16.
- Whitelock, Peter and Kieran Kilby (1995) *Linguistic and Computational Techniques in Machine Translation System Design* (Second edition). London: UCL Press.

CHAPTER 16

Post-editing

Jeffrey Allen

Mycom France, Paris, France

1. Introduction

This chapter describes the relevance, importance, and characteristics of MT post-editing. The task of post-editing has led to the creation of a new role, that of the **post-editor**, within the overall translation workflow process of many organizations that are implementing MT technologies. This discussion indicates the current status of post-editing as well as where it is moving within the general field of translation.

2. What is post-editing?

It is important to clarify that the term “post-editing” (also often written non-hyphenated as “postediting”) has been used within different subfields of natural language processing, including MT, automated error correction, optical character recognition, translation memory, and even controlled language. Post-editing is by far most commonly associated as a task related to MT and has been previously defined as the “term used for the correction of machine translation output by human linguists/editors” (Veale and Way, 1997). Another good summary statement indicates that “post-editing entails correction of a pre-translated text rather than translation ‘from scratch’” (Wagner, 1985). In basic terms, the task of the post-editor is to edit, modify and/or correct pre-translated text that has been processed by an MT system from a source language into (a) target language(s).

The inclusion of MT into translation and localization workflow processes has brought up a question that has never really been touched upon before in the field of traditional translation, referred to here as Human Translation (HT). This question concerns the acceptance and use of half-finished texts. Within the HT profession, creating half-finished texts is a non-issue because producing a partially completed

translated text is not something that human translators do. However, the primary concern for post-editing is that incorporating MT systems into the translation process results in creating a “raw” output translated text that is considered upfront to be a partially or incompletely finished text (also called “quasi-text”). It is therefore important to determine to what extent MT output texts are acceptable, and how much human effort is necessary to improve such imperfect texts. This human effort can be measured as the cognitive effort exerted to identify the corrections (especially since post-editing is a different task from translating or revising), as well as the manual effort to make the corrections on paper and/or on-line. Few benchmark tests have been conducted to estimate the productivity gain or loss of the post-editing process in comparison with the HT process. As of the writing of this chapter, the only tests that have been identified are those that have been conducted during pilot and production phases at Caterpillar Inc. (1995 onward), during the pilot phase at General Motors (summer 1999), and now currently being conducted at ABLE International (2000).

Pre-editing and controlled language writing principles are often used in tandem with the post-editing approach in order to improve the translatability of technical texts and to speed up the productivity of the post-editing process. We would like to state simply that it has been claimed that controlled language writing enhances and speeds up the translation and post-editing process (see Chapter 14).

3 Who are the post-editors?

The new role created by incorporating MT systems into the translation workflow process is the position referred to as the MT (translation) post-editor, or simply the post-editor. Attempts have been made in recent years to locate and temporarily or permanently hire post-editors from a pool of experienced HT translators, for example within the European Commission (EC)¹ and by ABLE International.² Since this role is so new to the field of translation, a limited number of methodologies have been and are currently being developed in several independent departments, institutes and companies on how to train post-editors to perform their post-editing tasks. Post-editors constantly struggle with the issue of the quantity of elements to change while also keeping the translated text at a sufficient level of quality.

Very few reports or talks are publicly available that describe the details and results of post-editing. For the most part, we are aware that the majority of translation/localization agencies and in-house translation departments that are conducting post-editing in production environments are creating their own sets of post-editing criteria, and in many cases are running the risk of re-inventing the same wheel. As a result of the increasing need for post-editors with respect to the

MT market, an initiative known as the Post-Editing Special Interest Group (see Allen, 1999) was set up by a few members of the Association for MT in the Americas (AMTA) and the European Association for MT (EAMT) to help plan and determine post-editing guidelines as well as propose a means of establishing a post-editing qualification program similar to the American Translators Association (ATA) certification tests.

The majority of existing experienced post-editors are mainly in-house staff translators (e.g., Caterpillar Inc., Pan-American Health Organization, EC Translation Service) and now a growing number of HT professionals who have been recruited as post-editing free-lancers through the EC and translation/localization companies such as ABLE International and the Detroit Translation Bureau.³

4. Reasons for using MT and post-editing

Post-editing is directly related to the integration and the implementation of MT systems. We note that there are several reasons that have led to the increased introduction of MT into the translation and localization scene over the past 10 to 15 years. In this section, we explain just a few of the many factors that have been pushing for the need to implement such translation technologies and have created the need for supplying the market with new post-editing skills.

One of the primary reasons for considering the use of MT, which incites the need for subsequent treatment of MT output texts by way of post-editing, comes from the constantly increasing focus on globalization (Dunlap, 1999). Large corporations and small- to medium-sized companies alike have been undergoing the expansion of their business to the four corners of the earth. It is no longer possible to rely on local business, nor to base one's commercial expectations on a single language as the sole medium of communication. The localization industry grew out of this globalization expansion. The result is that business can no longer be conducted just in English, or French, or Spanish or German. For companies to be successful on an ongoing basis, they must present themselves in a multilingual way. Looking back over the past 15 years, we see multiple industries which had traditionally only presented their information in English and which have moved toward a multilingual approach. As a result, this has created a ballooning effect with regard to the volume of translation jobs. In a short informal survey conducted by the present author among various well-known translation and localization agencies in mid-1998, it was found that all such agencies were experiencing an overall 30% annual increase in translation requests. Also, the EC has even claimed up to 50% increase in translation requests per year. It is also not uncommon that companies now receive translation requests for specific types of manuals and documentation that were

never previously translated. Many very successful software companies are now working in a simultaneous shipping (“simship”) mode whereby documentation in the source language plus a core group of target languages are all shipped at the same time. This requirement is putting significant constraints on the entire authoring, translation and localization process. Given recent statistics on the worldwide HT workforce,⁴ and that many companies and international corporations must provide their technical and marketing information in five, ten, twenty, and sometimes up to fifty or more languages, it is economically and nearly realistically impossible to meet these growing needs with such a limited HT workforce. With such an increased demand for translation, many companies are actively seeking ways to meet their translation needs within a reasonably affordable price range. Globalization and localization are significant factors that influence MT, and therefore the use of MT post-editing.

Another factor is the change in expectations with regard to the type and quality of translated material. Translation has traditionally been considered to be a customized process with the ultimate goal of a high-quality text product. In many cases, this still remains true, and is quite necessary. Highly sensitive documents, especially those containing information on user safety and security, obviously require a high-level treatment for translated version. The same is true for marketing information. However, there has been a steady increase with regard to the need for translation gisting, where users just want to understand in their own native language(s) the main idea(s) of a document that only exists in a foreign language. For such needs, a perfect translation and the ensuing details are not as critical. With the introduction of the *babelfish* translation portal on AltaVista and many other similar free translation portals⁵ that are now available via the Internet, the opportunity for translation gisting has become a very important means by which anyone can easily read potentially relevant and interesting information in any of the main international foreign languages (English, French, Spanish, Portuguese, German, Italian, Russian, Chinese, Japanese, and a few others) without having to learn these languages. Numerous reports by well-known survey and consultant organizations (Forrester Research, Ovum, Equipe Consortium, Allied Business Intelligence, International Data Corporation, Andersen Consulting, Bureau van Dijk) have appeared over the past year indicating that the expansion of the Internet in home environments is providing private consumers with access to information that these consumers have never been able to tap into before. With claims in reports by all of the above-mentioned organizations that around half of all Internet sites are in languages other than English, it is important to have processes in place that can provide for rapid gisting translations for readers to generally understand the information that is presented.

5. Types and levels of MT post-editing

The level of post-editing to be performed on a text is entirely dependent on several factors, including

- the user/client,
- the volume of documentation expected to be processed,
- the expectation with regard to the level of quality for reading the final draft of the translated product,
- the translation turn-around time,
- the use of the document with regard to the life expectancy and perishability of the information,
- the use of the final text in the range from information gisting to publishable information.

An entire additional book could be devoted to case studies treating the reasoning behind the combination of these and other factors. We cannot discuss here all such details. Each case study is different and should be recognized as such. We would like however to state that much consideration (as should be the case) often goes into the decision of choosing whether or not to integrate and use the MT and post-editing processes in a given environment.

As previously stated by Anne-Marie Loffler-Laurian, good evaluation criteria for post-editing would be best based upon an understanding of the objective of a given translated text (for example, gist reading vs. a published text to be disseminated.⁶ We therefore subcategorize in the following sections the types of post-editing level based on the different approaches to MT use for translation tasks.

The two main approaches for using MT systems are easily summed up as being either for inbound or outbound translation activities. Inbound translation (also referred to elsewhere as MT for acquisition or assimilation) is simply the process of translating to understand. Outbound translation (also referred to as MT for dissemination), on the other hand, the process of translating to communicate.

5.1 Inbound translation approach

There are several levels of use of texts and correction strategies within the inbound translation approach. They are described below.

5.1.1 MT with no post-editing (browsing/gisting)

Information translation **gisting** (also called “translation browsing”) is one of the primary motivating reasons today for having Internet MT portals⁷ available on the

Internet. This approach does not include any post-editing at all. It bypasses human intervention by presenting raw MT output text to readers, usually via the free Internet MT portals or else via enterprise intranet solutions, as a way of presenting a comprehensible translation of a foreign language text to readers in their mother tongue or in a language that they are proficient in. With inbound translation gisting, users themselves have control over the MT button for reading foreign-language texts. They acknowledge MT as a means to obtain valuable information that is unusable for them in the source language. Also, these users determine their own threshold acceptance of MT output. In today's multilingual superhighway of information, they choose to use MT as a way of gathering information that would otherwise be useless to them.

In essence, this browsing-gisting approach is of some value but is obviously not sufficient for all cases, hence the need for post-editing.

5.1.2 *Rapid post-editing*

The EC acquired specific licensing rights for the *Systran* MT system in 1976 and have been customizing the *Systran* EC version on-site ever since. The striking increase in MT usage by the EC's operating departments at the beginning of the 1990s indicated a specific need for dealing with urgent translations that could not be met by traditional translation channels. The EC's post-editing service was created as a response to providing rapid translation revisions of MT output. **Rapid post-editing** (RPE) thus came into existence to provide translations for urgent texts that are intended merely for information purposes or for restricted circulation, such as working papers for internal meetings, minutes of meetings, technical reports or annexes. In some cases, such documents are specifically classified as "for information only", as is the case for documents of similar nature within the United States Air Force.

In general, the main idea of RPE is to perform a strictly minimal amount of corrections on documents that usually contain perishable information (i.e., having a very short life span). Such documents, in essence, are not necessarily intended for public use, nor for wide circulation, etc. This is strictly minimal editing on texts in order to remove blatant and significant errors and therefore stylistic issues should not be considered. The objective is to provide the minimum amount of necessary correction work that can be made on a text in order for the text to be fully understandable as a comprehensible element of information. Although RPE has been used, explicit descriptions of RPE have not necessarily been established by organizations that use this post-editing method.

However, one of the main warnings that must be made, and which we are all certainly aware of in today's world of intranet network and e-mail communication, is that one can never be sure that the recipient of an RPE-processed document,

which may have initially been intended for internal use only, will not turn around and send or use it elsewhere. For example, how many times have you sent a private e-mail message to your boss or a colleague and found out later that portions of, or the entire text, were forwarded on to someone else? This is a major concern with regard to the new electronic age, and this is obviously a concern for the circulation of documents that undergo RPE processing.

5.2 Outbound translation approach

In contrast to inbound translation purposes, the outbound approach, and the levels of corresponding post-editing, aim at applying to a raw translation appropriate corrections for published documents that are destined to be read by many people.

In the past, this has often been referred to as “maximal post-editing”. The issue that has significantly hindered maximal post-editing is that with the high number of corrections that must be made for high-quality translated documentation, some highly-experienced human translators with excellent typing skills, or who have trained their speech dictation applications very well, can create the translated document from scratch nearly as fast as it would take to maximally post-edit an MT raw output version of it. We are even aware of translators who have conducted full post-editing for a period of time, and have abandoned working in such an environment because this can often be as or even more time-consuming than translating a target text from scratch. One of the main reasons for this is all too simply the working and interface environment. The abandonment of post-editing can happen when a functional and usable tool and methodology have not been made available. On the other hand, easy-to-use post-editing tools which are integrated into standard word-processing applications can be translation productivity boosters.

5.2.1 *MT with no post-editing*

It is first important to comment briefly on the possibility of 100% MT with no post-editing for the outbound translation approach. This notion was publicized in the 1980s and dwindled off during the 1990s once developers and implementers realized the incredibly complex issues involved in knowledge management, document processing, authoring/translation/localization within industrial and corporate contexts, etc. Upon implementation, the claims of 100% MT (with no post-editing) were modified to 90% MT accuracy (with 10% post-editing) for acceptance, and then subsequently modified to 80% MT accuracy (with 20% post-editing), etc. It is also important to note that differing percentages of MT accuracy have even been found when applied to different subdomains and different document types within the same technical domain. It is not possible to say that a given company will always achieve a specific percentage of MT accuracy and an exact complementary percentage of

post-editing, because all institutions normally process a variety of types of information and document types. It is thus more appropriate to say that an average X percentage level of post-editing has been attained in Y subdomain in Z company. After calculating the percentage of accuracy per subdomain, even within the same domain, an overall average can be presented. For example, when the present author worked on the Caterpillar Inc. MT project, we were confronted with a range of manuals (operator manual, service manual, diagnostic and troubleshooting manual, assembly and disassembly manual, etc.) that each represent different styles of writing. It was not unusual that the MT system we were using gave different output per document type, and thus required more or less post-editing corrections. Another example includes a more recent experience in adapting MT systems to deal with information ranging anywhere from that which is found in Web pages, in e-mail correspondence between different international branches of transnational companies, to content gisting of foreign-language information and documentation by employees and researchers in national organizations, to domain-specific applications for improving the productivity of translation teams. Such a range of information and documentation needs clearly shows how difficult it is to claim 100% accuracy of MT systems for published documentation.

For outbound published documentation, the only domain which up to this point has had very consistent and published results for non-post-edited or limited post-edited information is that regarding weather bulletins. Published results about the *Météo* system (see Chapter 15) have consistently demonstrated that it is possible to reach 90–95% MT accuracy; little, if no, post-editing is required. In all other cases, to our knowledge, especially for cases where documentation is published or used by third-party users, a minimal level of post-editing is necessary.

Given that MT without post-editing for outbound translation is limited in use and applicability for information dissemination activities, what are the different types or levels of post-editing for publishing documentation? Let us now look at the different types of post-editing and how they reflect the translation expectations.

5.2.2 *Minimal post-editing*

In the 1990s, the term “minimal post-editing” (sometimes also referred to as “post-editing at a minimum”) came into common use in the industrial and corporate sectors. Despite the multiple terms that are used for this concept, the main problem with minimal post-editing is how to quantify the amount of post-editing changes that must be made to a raw MT output text.

Minimal post-editing is a fuzzy, wide-range category because it often depends on how the post-editors define and implement the “minimum” amount of changes to make in view of the client/reader audience. Due to the fact that the resulting documents are almost always destined for distribution, whether this be internal or

external distribution with regard to the organization, the level of sensibility of interpreting the concept of “minimal” post-editing often seems to vary from one post-editor to another, from one manager to another, from one reviser to another. According to our investigations across different sectors over the past three years, specific post-editing guidelines are known to have been established within some organizations but certainly not within all organizations that are implementing post-editing. In most cases, there appears to be a missing link between the development of the systems and the training on how to use them and the resulting output. This is definitely an area which requires improvement for enhancing translation and post-editing productivity in the coming few years.

We must also take into account the psycho-social issues of the translation process. Editors and revisers in the authoring, translation, and publication sectors have often accumulated years of experience, something which is certainly not trivial by any means. Such experts are possibly plagued by the “red pen syndrome” which implies that any work-related document is subject to being edited with visible red ink, that the corrections should be made as quickly as possible, and higher levels of comment indicate higher productivity on the part of the editor/reviser who has reviewed the document. In essence, the same number of editing comments should be made either on a single reviewed document or distributed across multiple documents. In other words, documents destined for high-quality publication require high-quality editing in a workflow environment in which documents transit from junior translators to senior translators to translation revisers/editors. The editing process is therefore a learning process that often takes several years to fully comprehend and acquire under the tutelage of editor/reviser mentors who flood the junior staff members with an abundance of “red ink” comments.

Placing an MT system and a post-editing process into such a high-productivity environment is different from the typical translation editing workflow process. This is because the guiding objective in the minimal post-editing context is to make the least amount of comments possible for producing an understandable working document, rather than producing a high-quality document. This constraint can thus lead to one of two possible scenarios. The first case is that of over-correcting whereby the post-editor spends too much time on the post-editing process (also referred to as “over-engineering”, Godden, 1999), or secondly that of under-correcting whereby the post-editor does not sufficiently review the document and lets significant errors appear in the resulting final text.

The main issue with minimal post-editing environments is that there is often a large range of variation with regard to how post-editors interpret the level of corrections to be applied to the raw MT output texts. Also, although the objective is to make a minimal amount of corrective changes on any given document, this is more often than not negatively compensated by time-consuming **bug reports**,

which do not count toward user productivity, and which must be filled out by these same users and submitted to MT system developers. Thus, what is productive for the development team can be considered to be a loss in productivity for the production users. Professional translators should keep this in mind and thus negotiate the financial aspects of bug-reporting activities into their contracts with employees and clients.

A typical example of minimal post-editing is when the final document is expected to be sent out to third-party user/readers, even clients. Since the information will be disseminated, and thus published in some form or other, this implies that minimal post-editing must be applied to the MT output texts before passing these texts on to the third-party users. Many reports simply mention post-editing in a dissemination context whereby partial or minimal post-editing is the main objective. Those articles and reports discuss the general concept and ideas, but very few public reports give any concrete data with regard to what constitute the specific criteria of partial or minimal post-editing, the linguistic revision categories, the quality assurance metrics employed, etc. Several unpublished internal technical reports have been written that describe such guidelines and for which examples cannot be specifically cited in this chapter.

5.2.3 Full post-editing

The idea of full post-editing of texts has been debated for many years because full post-editing implies a high level of quality of the resulting texts. The question is based on the notion of whether it would be faster to post-edit the raw MT output or simply translate the document from scratch. It has been shown by specific industrial projects that post-editing on documents written according to controlled language principles takes less time than translating the entire document without any computer-aided translation assistance. The use of full post-editing on uncontrolled input language texts has generally been avoided in the past. However, recent activities by localization and translation agencies (e.g., ABLE International) that use MT systems for translating texts without following any specific controlled input grammar or writing guidelines, indicate that a market for full post-editing may in fact be underway. Only time will tell with regard to the productivity that can be attained in such contexts.

6. Post-editing guidelines and criteria

It is fine to talk about the idea of having levels of post-editing, but what most people really want to know is what are the actual post-editing principles or guidelines that support the post-editing concept. As said earlier, it is often claimed that post-

editing is efficient, that it is faster, etc., but very little concrete data has been made available with regard to user studies, results and the methodology employed for post-editing. We provide below a number of short case studies as well as post-editing principles and examples that have been collected from published documents. There are several other companies that have created post-editing principles but have not released them beyond internal technical reports, so these examples cannot be cited in this chapter. It is also important to note that a few other companies are new to the area of post-editing activities and have been recently (during the year 2000) involved in developing post-editing guidelines for their in-house staff and external free-lance post-editing teams. Due to this new and rapidly expanding area of interest, the most recent sets of post-editing guidelines by these institutions are not yet available for public sharing at the time of the writing of this chapter.

6.1 General Motors and SAE J2450

One of the most concrete cases of establishing and using documentation for post-editing is on the CASL (Controlled Automotive Service Language) project at General Motors. CASL minimal post-editing uses the Society for Automotive Engineering (SAE) J2450 standard metric for translation quality.⁸ The SAE J2450 working group⁹ has developed a standard that specifies several categories of errors which are rated as unacceptable in translated texts; this standard however does not address stylistic considerations within any of the error categories. During the post-editing process, the post-editor is simply requested to identify and correct all occurrences of J2450-type errors that are discovered in raw MT output text. J2450 also provides weights for each type of error, subcategorized into distinct levels of serious and minor errors: there is therefore an objective means of calculating a final score for post-editing processing that has been conducted on a given text.

The order of priority of errors according to the J2450 standard is listed below:

- A. Wrong term (WT)
- B. Syntactic error (SE)
- C. Omission (OM)
- D. Word-structure or agreement error (SA)
- E. Misspelling (SP)
- F. Punctuation error (PE)
- G. Miscellaneous error (ME)

The J2450 metric is implemented by translation suppliers of General Motors, including its external translation suppliers which provide post-editing-type translations.

6.2 Pan-American Health Organization (PAHO)

Muriel Vasconcellos, the former Chief of the Terminology and Machine Translation Program at the Pan-American Health Organization (PAHO), has written many articles on the topic of post-editing from experience of herself and fellow post-editing colleagues who have used PAHO's Spanish–English *Spanam* MT system. In Vasconcellos (1986), she address a number of specific issues that had to be addressed in the post-editing process. In that article, she indicates the following points that are specific to Spanish and that have required significant post-editing correction work for texts that have been submitted to the *Spanam* system because of insufficient analysis by that MT system at that time:

- Verb+*se* as theme,
- Adjunct theme (cognitive) followed by verb+*se*,
- Adjunct theme (non-cognitive) followed by verb+*se*,
- Verb+*se* after a dependent clause,
- Embedded verb+*se*,
- Fronted verb in embedded clause,
- Participial theme.

A few other general comments in that article are the following:

- verb fronting in Spanish is translated by *Spanam* with dummy subjects (impersonal subjects), yet is not appropriate,
- thematic verb with postposed subject,
- problem of participial clauses with postposed subject nominal that exists in Spanish but has no equivalent in English,
- infinitival clauses in Spanish (nominalization effect in Spanish) but not in English.

The above-mentioned points are all specific to conflicts in language typologies between Spanish and English. Numerous examples of each point are given in the original article.

As of first quarter 2000, Marjorie León of PAHO indicated that their unit does not provide any formal training for their post-editors. Post-editors are rather provided with a set of post-editing “macros” and some basic guidance about how to take advantage of the raw MT output text, how to avoid extensive reordering of concepts, how to respect phrases that are enclosed in “reliability marks” in the output, how to deal with context-sensitive alternate translations, etc. Although formal post-editing training is not necessarily provided, PAHO post-editors are informed when they change too much in the post-editing process or if they fail to correct essential elements.

6.3 Loffler-Laurian (1996)

A book that is well worth investing in, for readers of French, is Loffler-Laurian's (1996) book on the topic of MT. It is however important to note that this book is based on research conducted on MT systems and post-editing during the 1980s. Thus, some claims about frequent MT errors requiring post-editing (numbers, digits in expressions for dates, measures, quantities, currencies, headings and sub-headings, and percentages cited on page 56) have in many cases been partially or fully resolved by a combination of new computer-user environments and enhanced translation systems that have been developed since that time. Many newer versions of MT systems are now compatible with the most commonly used operating systems (Windows 3.x, 95, 98, 2000, NT, Macintosh OS 7–9, Unix) and now some even integrate fully into the most standard desktop publishing software (Microsoft Word, Excel, Powerpoint, Corel WordPerfect, Framemaker, Interleaf, etc).

Loffler-Laurian also indicates that the ability of an MT system to preserve formatting of the original text document is important since the reformatting of a post-edited document transferred to a standard word-processor results in additional wasted time. This problem has been dealt with by the fact that nearly all commercial MT systems that integrate into standard word-processing packages and into web browsers must deal with how to maintain page formatting. Formatting is now maintained, along with paragraph alignment (in either horizontal or vertical dual-window display) and color-coded identification markers. These technical improvements since the mid-1990s in both custom-made MT applications and in commercially distributed MT software packages have thus led to a decrease in post-editing time for dealing with the more or less non-linguistic issue of formatting (Allen, 2002b). We do note that one type of formatting feature that is not always present in commercial systems is the ability to copy and paste text from a word-processing application (e.g., Microsoft Word) directly into the MT software interface without creating truncated sentences due to hard carriage returns. Although a few MT products still produce this problem, which obviously results in more post-editing than is necessary, it is an issue that is on the top of the priority list for MT companies that must sell their products to the general public and to corporate clients.

Also, punctuation (p. 58) may have been a problematic issue for MT output at the time of the writing of her book. However, enhanced algorithms have been developed over the past few years to provide for more robust MT processing of punctuation in multiple languages. This is a significant issue for research and development MT teams in both commercial and academic MT research circles. The current conversion process is by no means 100% perfect, but the identification capabilities have come quite far and should no longer be considered as a major error for post-editing.

After having discounted the numerous conversion and formatting issues that required much work of post-editors in the past, we can now come to the linguistic issues that continue to perturb post-editors in their work. Loffler-Laurian sets forth (pp. 93–94) a number of general criteria that should be followed in order for a post-editor to be most effective and efficient during the post-editing process.

1. criterion of situation and document type;
2. criterion of necessity;
3. criterion of comprehensibility.

The criterion of **situation and document type** refers to the extralinguistic constraints that relate to the document. This is important because the requirements for post-editing a procedural text might be different from those of a heavily descriptive text on electronic troubleshooting. Also, the objectives of the writer and reader audience, the eventual distribution of the document, and other specific expectations are issues that must be clearly communicated to the post-editor in order for the task to be completed according to the stated requirements.

The criterion of **necessity** is very well stated by Loffler-Laurian on page 94: “One runs into … the question of the line between decor and necessity, correction and convenience, clarity and beauty.”¹⁰ Basically, it is a constant juggling game between what is minimally necessary and what is added to have a slightly better text.

The criterion of **comprehensibility** is one which everyone refers to, but few have clarified in writing through user studies. As stated by Loffler-Laurian (*idem.*), “This criterion needs to be investigated in more depth with studies even on the concept of comprehensibility and the threshold level of comprehensibility.”¹¹ Several studies along these lines have been conducted on controlled language and could be extended into the related area of post-editing.¹²

One last point to be addressed, which has been mentioned by Loffler-Laurian as a problem for post-editing, is the issue of lexical ambiguity. Lexical ambiguity has always been a problem for MT systems. The first case is that of words that have two or more meanings within the same grammatical category. For example, the English word *fall* can be understood as the autumn season of the year as well as the result of descending rapidly. In many languages, this noun can be translated by two different words. Another case of ambiguity mentioned by Loffler-Laurian (p. 58) concerns grammatical homographs that are words having the same spelled form but having different meanings or playing different grammatical roles in a sentence. An example of this can be found with the English word *running* that is a gerund noun (as in *running is a type of exercise*), a present progressive verb (as in *I am running*), and an adjective (as in *Measure the revolutions of the running engine*). Grammatical homographs can be dealt with in “Professional” and “Corporate” versions of commercial MT systems that allow translators and post-editing specialists to create

their own customized dictionaries and thus immediately influence and improve the MT output. With proper training on MT dictionary creation techniques, it is possible to circumvent the ambiguity to a certain extent. One word of caution, however, is to obtain such training from experts who have much experience in MT dictionary building methodology. This allows the users to avoid over-engineering their dictionary and to reduce the risk of entries that could create other unforeseen translation problems for the MT system.

6.4 European Commission Translation Service (ECTS)

Some of the only post-editing guidelines provided by the ECTS were written by Emma Wagner (1985). Many of the guidelines given in her set of post-editing working procedures actually focused on keeping in the mind the need to complete post-edited texts with heavy turn-around time constraints. For example:

- Do retain as much of the raw translation as possible. Resist the temptation to delete and rewrite too much. Remember that many of the words you need are there somewhere, but probably in the wrong order.
- Don't allow yourself to hesitate too long over any particular problem — put in a marker and go back to the problem later if necessary.
- Don't worry if the style of the translation is repetitive or pedestrian — there is no need to change words simply for the sake of elegant variation.
- Don't embark on time-consuming research. Use only rapid research aids (*Eurodicautom*, knowledgeable colleagues, specialised terminology lists — which can be stored on the word processor and accessed directly if you work on screen). If a terminology problem is insoluble, bring it to the attention of the requester by putting a question mark in the margin.

Above and beyond the time-constraint mind-set aid provided by Wagner, her guidelines also provided a minimal set of linguistic criteria that cover all language directions:

- Do make changes only when they are absolutely necessary, i.e. correct only words or phrases that are
 - a) nonsensical
 - b) wrong
 - and, if there is enough time left,
 - c) ambiguous.

The guidelines provided by Wagner show that a methodology of post-editing tends to be primarily based on the philosophy of dealing with turn-around time constraints more than on pure linguistic changes that are to be made during the post-editing process.

Then, due to a constantly increasing demand of post-editing requests during the early 1990s that have been filled by the ECTS, a call for tenders for post-editing services through the ECTS was concluded in 1998.¹³ Post-editors are required to carry out rapid revision of output generated by the EC's MT system in combinations of English, French and German.¹⁴ An explanation of the post-editing service can be found in a few recent articles on this topic (e.g. Senez, 1998).

The ECTS-RPE unit aims at providing a service that follows the following three stipulations:

1. the customer urgently needs a version of the text in another language;
2. the text is not destined for publication, but will serve some temporary purpose;
3. the customer is fully aware of the process involved in producing the post-edited text.

As of March 1990, the ECTS-RPE unit led by Dorothy Senez does not have a formal post-editing training guide or specific guidelines for post-editors. The internal EC post-editors are those who have already received on-the-job experience, and the external post-editing vendors were selected according to proof of experience in MT post-editing. It appears that a choice was made to select experienced post-editors rather than try to train an entire new team from scratch by creating training programs and course sets which can be time-consuming to develop and implement.

6.5 Post-Editing Special Interest Group

As briefly mentioned above, a Post-Editing SIG was set up by a few members of AMTA. This group met at the AMTA-98 meeting in Langhorne, Pennsylvania, then at the Third International Controlled Language Applications Workshop (CLAW 2000), in Seattle, Washington, and at AMTA-2000 in Cuernavaca, Mexico.

The main thrusts for the SIG include:

- developing specifications for what would be an optimum post-editing environment;
- educating the various audiences which need to know more about post-editing;
- promoting post-editing workshops at conferences that are close to the professional translation community;
- developing post-editing courseware for translation programs.

7. Semi-automating Post-editing Processing

It is obvious from all of the preceding sections that a list of specific post-editing criteria and guidelines is very difficult to locate. There is a potential for much variation in post-editing guidelines, especially given the different language directions that are possible. Another risk is the ongoing reinvention of such post-editing principles across different organizations. And thirdly, much energy can be wasted on (re)creating principles to tell post-editors to fix up the highly frequent, small MT raw output mistakes that unnecessarily add to the cognitive load on these experienced language experts. Since the ECTS-RPE unit had no formal post-editing training or specific guidelines for post-editors, Jeffrey Allen and Christopher Hogan took the initiative to contact the ECTS by proposing a specific research-oriented task of developing an automated post-editing (APE) module that is based on EC texts (Allen and Hogan, 2000).

The inspiration behind this development work of an APE module is that if an MT system makes a particular error when translating a document, it is very likely to commit the same error each and every time the same set of conditions are presented. And if the error is fixed in a similar way, then it is possible to capture these modifications and to implement them automatically so that such repetitive errors can be reduced in MT output.

Upon the presentation of a viable proof of concept for automatic APE in February 1999, this project obtained authorization in April 1999 from the ECTS to conduct experimental research on tri-text sets (source text, raw MT output, post-edited version) taken from the ECTS database; the only condition was that the selected files had to be pre-checked by the ECTS personnel for confidentiality. An initial test group of 8 sets of English to French tri-text files and 17 sets of French to English tri-text files were provided by the ECTS for the first stage of this research project. The ultimate objective of the project is to demonstrate that it is possible to implement new machine-assisted human translation (MAHT) solutions for production environments where MT systems are used, and to increase the production turn-around time with such tools.

More importantly, the systematic errors committed by an MT system recur not only within documents, but also across documents, and over time. Thus, without any semi-automated assistance, a post-editor is likely to have to fix the same errors again and again in daily post-editing tasks. The situation may be compared with that of the translator. A translator is likely to notice and remember the same phrases presented time and again in the translation process. For post-editing, it would be desirable to have a processing engine that could automatically fix up the highly frequent, repetitive errors in raw MT output before such texts are even given to a human post-editor. Such an APE module can speed up tasks of human post-editors

by eliminating many of the numerous but trivial replacements that are necessary to perform their job. In order to post-edit MT output automatically, the APE system automatically learns from previously post-edited documents, a process which has been tested and deemed successful from the generous donation of textual corpora provided by the ECTS for this project.

Tests conducted on the post-edited tri-texts provided by the ECTS have resulted in frequent grammatical constructions that have been learned by the APE module. Figure 1 shows a list of the most frequently occurring changes that the APE module learned from the ECTS texts.

the -> Ø / of _	(<i>the</i> is deleted after <i>of</i>)
the -> Ø / and _	(<i>the</i> is deleted after <i>and</i>)
information -> informing / _	(<i>information</i> is changed to <i>informing</i>)
at the time of -> At / _	(<i>at the time of</i> is changed to <i>at</i>)
to -> for / Commissioner _	(<i>Commissioner to</i> is changed to <i>Commissioner for</i>)
!!! Raw Machine Translation !!! -> Ø / _ ¹⁵	

Figure 1. Changes to ECTS texts learned by the APE module

The APE module is used to identify the most frequent constructions, such as those indicated above, and to learn the corrected forms from the post-edited versions of the tri-texts. By doing this, it is possible to learn automatically what the human post-editors have applied to texts. In a case where no post-editing criteria have been previously provided to the post-editors, this tool allows us to develop a set of naturally inherent post-editing rules for a given language direction and a given MT system. In the case where post-editing criteria have in fact been provided to the post-editors, it is possible to use the APE module as an evaluation tool for the implementation of such criteria in a translation production environment.

These initial results of an APE module have been obtained from less than 30 sets of tri-text files that were provided by the ECTS. Additional analyses will be conducted and made available in future studies based on a significantly larger set of files (about 200) that have recently been donated as a means to further improve the coverage of this machine learning APE method and to test its implementation.

For the most part, we are aware that nearly all translation/localization agencies and in-house translation departments that are conducting post-editing in production environments are often each creating their own sets of post-editing criteria. This seems to us to be reinventing the wheel. In some cases, there is not formal post-editing training, but rather an expectation that experienced, or even novice translators, when given a few basic guidelines, can devise their own post-editing rules and adhere to them through practice.

The APE, even in its prototype form, can be considered as a first-level productivity enhancement tool. It basically allows for the semi-automatic correction of the most common repetitive errors in raw MT output, thus letting the post-editors focus on the more essential changes. This tool is in no way designed to be a replacement option for human post-editors.

8. Conclusion

In this chapter, we have seen the most up-to-date information concerning MT post-editing and how it fits within current translation and localization processes. Although much information about post-editing user studies is company-specific and proprietary, that which has been made available for public use has been discussed herein. For cases where research and implementation departments have provided information about post-editing principles, this has been given. The development of a new automated post-editing module, APE, has been discussed as well. As seen from market activity among many new companies that are undertaking post-editing, it is expected that much more information will become available on this topic in the coming 2–3 years and will lead to further research and work.

At the time of final review of the present book, a newly published book on MT post-editing had appeared (Krings, 2001) with the goal of providing an objective and empirically based evaluation of MT and post-editing while taking into consideration the psycholinguistic point of view of the cost and effort of the latter. A comprehensive review of the book is also available (Allen, 2002a).

Notes

1. Information on the outcome of the tendering procedure for post-editing services can be found in the *Official Journal of the European Commission* No. S 204 (p. 40) of 21 October 1998.
2. Call for tenders for post-editors in the on-line LINGUIST List 10.1258, dated August 29th, 1999, <http://linguistlist.org/issues/10/10-1258.html>.
3. See www.ableintl.com and www.dtbonline.com.
4. Figures from Bureau van Dijk Electronic Publishers and Allied Business Intelligence cited in *Language International* 11.3, June 1999, pp. 19–20.
5. See Allen (2000), Bennett (2000), and Chapter 12.
6. “L’adaptation d’une traduction à son objectif (par exemple, lecture rapide pour information ou publication pour diffusion) serait un bon critère d’évaluation”, Loffler-Laurian (1996: 69).

7. See Allen (2000) and Gerber (2000). Also available on-line at www.eamt.org/resources/.
8. "Translation Quality Evaluation", *International Journal for Language and Documentation* 3, January 2000, p. 25.
9. The J2405 Task Force is a subcommittee to the SAE E/E Diagnostic Systems Committee and includes participation from General Motors, Ford, DaimlerChrysler, Volvo and their translation suppliers. See www.sae.org.
10. "On se heurte ... à la question de la ligne de démarcation entre décoration et nécessité, correction et convenance, clarté et beauté."
11. "Ce critère devrait être encore approfondi et des études restent à faire sur la notion même de compréhensibilité et du niveau-seuil de compréhensibilité."
12. Shubert et al. (1995), Chervak et al. (1996).
13. Information is available at <http://europa.eu.int/comm/translation/free-lance/en/ao-en.html>.
14. See also footnote 4.
15. The phrase "!!! Raw Machine Translation !!!" appears in all ECTS target-language files of MT output as a warning to readers. This phrase is removed in the post-edited versions once the text has been reviewed by a post-editor.

References

- Allen, Jeffrey (1999) "SIG on MT Post-Editing Established at AMTA-98", *MT News International*, 21, February 1999, page 13.
- Allen, Jeffrey (2000) "The Value of Internet Translation Portals", *International Journal for Language and Documentation*, 2000-6, 45-46.
- Allen, Jeffrey (2002a) "Review of Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes", *Multilingual Computing and Technology* 13.2, 27-29.
- Allen, Jeffey (2002b) "Review of Reverso Pro 5 and Reverso Expert", *Multilingual Computing and Technology* 13.6, 18-21.
- Allen, Jeffrey and Christopher Hogan (2000) "Toward the Development of a Postediting Module for Raw Machine Translation Output: A Controlled Language Perspective", in *Proceedings of the Third International Controlled Language Applications CLAW 2000*, Seattle, Washington, pages 62-71.
- Bennett, Winfield Scott (2000) "Taking the Babble out of Babel Fish", *Language International* 12.3 (Special issue on Machine Translation), 20-21.
- Chervak, Steve, Colin Drury, and James Ouellette (1996) "Field Evaluation of Simplified English for Aircraft Workcards," in *Proceedings of the 10th FAA/AAM Meeting on Human Factors in Aviation Maintenance and Inspection*, Alexandria, Virginia.
- Dunlap, Bill (1999) "Online Targeting by Language", *HLT News* (online journal), www2.hltcentral.org/lejournal/article.asp?articleIndex=1571.
- Gerber, Laurie (2000) "Laurie's Links", *MT News International* 25, 13-14.

- Godden, Kurt (1999) “CASL: General Motors’ Controlled Language and Machine Translation Project”, presented at the First Multilingual Documentation TopTec Symposium for the Automotive Industry, Amsterdam, 21st October 1999.
- Krings, Hans (2001) *Repairing Texts: Empirical Investigations of Machine Translation Post-Editing Processes*, edited by Geoffrey Koby. Translated from German to English by Geoffrey Koby, Gregory Shreve, Katjz Mischerikow and Sarah Litzer. Ohio: Kent State University Press.
- Loffler-Laurian, Anne-Marie (1996) *La Traduction Automatique*. Villeneuve d'Ascq (Nord): Presses Universitaires du Septentrion.
- Senez, Dorothy (1998) “The Machine Translation Help Desk and the Postediting Service”, *Terminologie & Traduction* 1998-1, 289–295.
- Shubert, Serene, Jan Spyridakis, Heather Holmback, and M. B. Coney (1995) “The Comprehensibility of Simplified English in Procedures”, *Journal of Technical Writing and Communication* 25, 347–336.
- Vasconcellos, Muriel (1986) “Functional Considerations in the Postediting of Machine-Translated Output”, *Computers and Translation* 1, 21–38.
- Veale, Tony & Andy Way (1997) “*Gaijin*: A Bootstrapping Approach to Example-Based Machine Translation”, in *International Conference, Recent Advances in Natural Language Processing*, Tsigov Chark, Bulgaria, pages 239–244; also available at www.compapp.dcu.ie/~tonyv/papers/gaijin.html.
- Wagner, Emma (1985) “Post-editing Systran — A Challenge for Commission Translators”, *Terminologie & Traduction* 1985-3.

CHAPTER 17

Machine translation in the classroom

Harold Somers
UMIST, Manchester, England

1. Introduction

The use of MT and related software¹ in the classroom has various perspectives depending on the type of “student”: one is teaching about computers and translation for its own sake, as part of course in one of the contributing fields such as linguistics, computational linguistics, computer science, information technology and so on. We will have only a little to say about this viewpoint, since it is presumably not the main interest of the reader of this book (though it may still be of interest). Another, most closely related to the theme of this book, is teaching trainee translators and other professional linguists about translation software. A third is the role (if any) of this software for teaching languages. Finally, end-users provide a further perspective on the question of how to teach MT. In this chapter we will try to synthesize and expand on these disparate views.

In the next section we discuss why translators and other language professionals should know about translation software, and make some suggestions about the way the subject can be presented. In a sense, this entire book can be seen as a kind of textbook for this activity, and so we concentrate on practical aspects of familiarizing trainee translators with translation software and related themes.

In Section 3 we shift the focus to the possible use of translation software in language teaching. There is an obvious overlap inasmuch as some (though by no means all) language teaching is related to translation as a linguistic activity, but we will also review some proposals for the use of translation software to enhance language learners’ perception of contrastive differences between languages, and to help them learn aspects of second-language grammar and syntax.

One possible set of “targets” for teaching MT are end-users. In Section 4 we consider some strategies for educating users such as scientists and business people into the best uses of MT. Included here are some comments on addressing the problem of “language-pair deficiency”, when there is no appropriate software available for a language-pair that interests a particular student.

In the final section we will look briefly at the aspects of MT that might interest linguists, computational linguists, computer scientists and so on, and we will suggest some ways in which translation software can be used to illustrate these areas of interest.

2. Teaching trainee translators about MT

Of most relevance to the readership of this book, we would claim that trainee translators and other professional linguists need to understand what translation software can and, perhaps even more important, *cannot* do. Indeed, much of this book is aimed at that exact goal. We have tried to give some insight into how translation software works, why it is difficult, what kind of translation tasks such software is appropriate for, what alternative computational tools are available and how to integrate them into the workflow.

In fact, in some countries, awareness of the role of the computer has long had a recognised place in translator training: the German BDÜ (*Bundesverband der Dolmetscher und Übersetzer*)² promoted it in 1986 and various studies were conducted in the 1990s,³ culminating in the 15-month LETRAC (Language Engineering for Translators' Curricula) project funded by the European Commission in 1999, which investigated the inclusion of information technology in a number of translator training courses in Europe.⁴

Hands-on experience of various tools is certainly an essential aspect of a translator's education. In the past, the expense of translation software has made it difficult for translator-training establishments to invest in software: pricing is more oriented towards professional users, though our experience is that discounts can be negotiated with some vendors for educational establishments. More recently the cost of translation software has fallen dramatically, and — assuming that computer labs to install the software are already available — obtaining a few systems for students to experiment with is quite a reasonable goal. Indeed, some students may be sufficiently impressed by the functionality of some translation software that they will buy their own copies of it. We have found it useful to obtain a range of software, including systems which we know to be among the less impressive: illustrating how bad translation software can be is a useful precursor to showing the best that it can offer. And one should also aim to illustrate the numerous computational aids for translators, as described in Chapters 2 and 3.

Students can of course be invited simply to familiarize themselves with the available software, perhaps being asked to do exercises which simulate a real-life translation situation. Trainee translators often have negative preconceptions about MT, and a useful initial exercise, suggested by Pérez-Ortiz and Forcada (2001),

involves using an MT system to translate a sentence first word by word and then as a whole sentence: the two translations will inevitably be different, and a discussion of the differences can help to underline the degree to which MT systems do encode at least some linguistic ability and sophistication. Building on this, one can then use specially designed exercises which expose students to the weaknesses of the software. We give here some examples of “trick” sentences that we have used to show some of the subtleties of natural language and how difficult these can be for computers. Even the best translation software packages will generally have some difficulties with some or all of the following. We show sentences for translation from French and German into English: mostly they cover the same linguistic problems, though one or two are particular to those language pairs. We invite the reader to try out these sentences on a translation software system (for example, AltaVista’s *babelfish*, available free on the Web) and decide for themselves what the problem is. Some suggested answers are given at the end of the chapter.

Take the attached set of example sentences in French or German, and use translation software to translate them.

You will find that for most of them, the output is unsatisfactory. For each example, state briefly why the translation is bad, by which is meant (a) what a better translation would be and (b) why it is difficult for the computer to achieve this.

If, in any cases, you think the output is good enough, say why you think the sentence might have been problematic.

French examples:

1. L’oiseau entra dans la chambre. L’oiseau entra dans la chambre en sautillant.
2. Charles se suicida.
3. On a donné le livre à Paul. On a dormi dans ce lit.
4. Nous venons de finir de lire ce livre.
5. Mon cousin est beau. Ma cousine est belle. Ma cousine est riche.
6. Les pieds de la table sont très épais.
7. J’ai loué la voiture de chez Avis. Avis m’a loué la voiture.
8. Le voleur donnait un coup de pied au gendarme. Le voleur donnait des coups violents de pied et de poing au gendarme.
9. Le pilote ferme la porte. Le pilote agile le porte.
10. Vous pouvez faire des achats de votre domicile.
11. Mon ancien mari a visité une ruine ancienne.

German examples:

1. Ich habe Hunger. Ich habe grossen Hunger.
2. Ich esse gern. Fritz spielt oft gern Tennis.
3. Das Mädchen gefällt dem Mann. Das Mädchen scheint, dem Mann zu gefallen.
4. Es wird getanzt und gegessen.
5. Hans will, dass Kurt sein Frühstück isst.

6. Ich liebe Kreuzworträtsel. Hans ist ein schneller Kreuzworträtsellöser.
7. Der Ladendiebstahl ist hier ein wichtiges Problem.
8. Meine Armbanduhr geht vor.
9. Der ehemalige Kanzler heisst Kohl. Herr Kohl ist jetzt im Ruhestand.
10. Die Tauben lassen die Gebäude in der Stadtmitte ganz schmutzig. Die Taube hat den Olivenzweig zurückgebracht.
11. Mein Vetter ist schön. Meine Kusine ist schön. Meine Kusine ist reich.
12. In dieser Universität studieren 3 000 Studenten und Studentinnen.

A number of other assignments and projects have been used by the present author with trainee translators studying MT, as detailed in the following sections.

2.1 Evaluation

Students can be asked to conduct a small-scale **evaluation** of the software, along the lines of evaluations described in Chapter 13. Depending on the time and effort that students are expected to put into this assignment, the evaluation can be more or less sophisticated. For most of the evaluations suggested in the literature, students have neither the time nor the resources to get statistically significant results. For example, any evaluation that requires judges to give a subjective evaluation of some aspect of the system requires quite a large population of judges. Nevertheless, they can gain a realistic impression of what is involved in setting up an evaluation even if they cannot see it through to its end result. **Comparative evaluation** of a single system translating different types of texts, or different systems translating the same text may be particularly revealing.

2.2 Post-editing

Post-editing to turn raw output into publishable quality is another exercise that students can undertake (cf. Chapter 16). O'Brien (2002) discusses teaching post-editing as an explicit skill needed by trainee translators, perhaps in connection with controlled language (cf. Chapter 14, and next section). Students can be given a text with or without its translation, or asked to find one themselves. They should work into their native language if possible, though this of course may not always be possible. This exercise can be given as a pure post-editing exercise, or students can be asked to comment on the problem, using the given text as a case study. Students could even be asked to formulate post-editing guidelines based on a certain piece of translation software: this involves first familiarizing themselves with the typical output of the system with a given type of text before drafting the guidelines.

2.3 Guidelines for controlled language

A similar exercise involves drafting controlled-language guidelines for use of a given system (see Chapter 14). Again, students should first get familiar with the behaviour of the system, and then develop a list of do's and don'ts that will promote good quality translation, and avoid the main pitfalls.

2.4 Dictionary updating

An important feature of most piece of systems is the ability to add items to the system dictionaries. This suggests a number of possible exercises and assignments. One way to do this is to give students a raw translation and an improved version (*not* post-edited) which is achievable by editing the system's dictionaries (this requires preparation on the part of the teacher of course), then ask the students to figure out how to edit the dictionaries so as to achieve the given target text.

More generally, students can be asked to evaluate different aspects of the dictionary updating procedures, in particular **how easy** this is in general, what effect it has, and how **effective** it is. The (perhaps subtle) difference between these last two is that “effect” is concerned with what the details in the dictionary relate to, and the “effectiveness” is whether changing the dictionaries does actually have the intended effect. For example, one system that we are familiar with invites users to stipulate a number of “translation attributes” when entering a new noun in the dictionary, as shown in Figure 1 (cf. Chapter 2, Figure 5).

One could evaluate the effect of these attributes by setting up a test suite of sentences, changing particular attributes and seeing whether the translation changes. This is a kind of “reverse engineering”, because we are trying to see how the system uses the information it asks us to give it.

Evaluating “effectiveness” tackles the problem from the other end, so to speak. In this case, we might have a certain effect in mind, and some assumption about how to achieve it. For example, in the documentation there might be some guidelines on how to get a certain result. A case in point comes again from the system illustrated above: when entering compound nouns such as French *poste de travail* ‘workstation’, the user should indicate with an asterisk which word(s) should be inflected for plural (see again Chapter 2, Figure 5). An effectiveness evaluation would confirm that marking *poste* as inflectable does indeed lead to the correct translations of both singular and plural, in both directions. This is of course a trivial example, but gives some idea of the kind of exercise that can be undertaken.

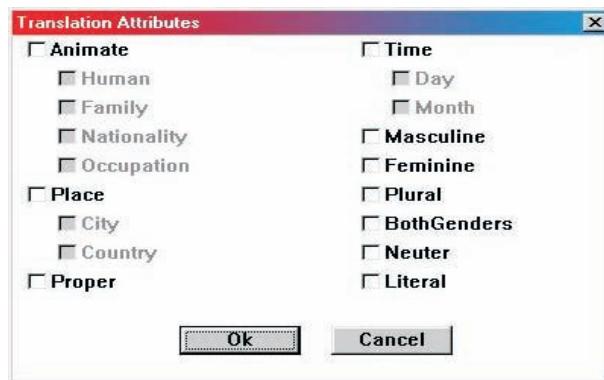


Figure 1. Semantic attributes for new dictionary entry⁵

2.5 Simulation of workflow scenarios

One of the aims of translator training is to give students competence and experience to carry out work in a professional manner, and to enable them to take a strategic view of the use of translation software within a particular organisational setting.⁶ Students need to be engaged in authentic tasks like those they will face as professionals in the field, and to be encouraged to reflect on the wider implications of any decisions they take regarding the use of translation software of any kind in carrying out these tasks. Studying **workflow scenarios** is an important aspect of translator training, and involves not only considering how tasks can be handled but also being able to explain and discuss the relative merits of possible alternatives.

Of course, in reality workflow involves clients; in a training situation, we can only simulate clients' attributes and requirements. But a typical translation workflow is not just a bi-directional relationship between the translator and the customer. Other parties involved might include revisers, brokers, publishers, the original author, even model end-users. Use of translation software might involve yet further contributors to the process: technicians, lexicographers, terminologists, and so on. Communication and exchange of data with these colleagues may be part of the scenario, and if they are located remotely, e-mail plays an important role.

Workflow scenarios can also involve a variety of resources. For example, if you were translating a brochure for a museum, it would be almost essential to actually visit the museum. Thinking of scenarios which have a particularly computational element, we can mention the case of receipt and despatch of work in machine-readable format (on floppy disk through the post, or as a word-processing file sent electronically), preparation of text for desk-top publishing, and, especially, com-

puter-relevant translation activities such as software localisation (see Chapter 5) and web-page translation. Another computational aspect to translation is the use of the World Wide Web as a source of background information: taking the case of the museum again, if you cannot visit it, perhaps a web site with photos is the next best thing.

Some scenarios have a longer-term perspective. The use of customized terminology databases and translation memories with regular clients is an obvious case in point. But one should not rule out also the possibility of introducing the client to controlled language, and working with them to develop a set of guidelines that might make use of translation software more attractive and feasible.

2.6 Criticism of documentation and general usability

Finally, students can be asked in a much more general way to evaluate the software, focussing on the relation between the documentation and the system itself, or, least specifically of all, giving their impressions of how usable the system is in general, in the manner of a software review, perhaps for a magazine or journal.

Reviewing the documentation in particular is an interesting exercise with many facets. Mowatt and Somers (2000) have developed a number of criteria for this kind of approach. They suggest evaluating whether the documentation is pitched at an **appropriate level** for the assumed users of the software, taking into consideration (i) the competence of the *typical user* in various areas, (ii) the competence *stated as being necessary* by the documentation and (iii) the competence *actually needed* to understand the documentation. The **quality** of the documentation can also be assessed by looking at the complexity of the language, the appropriateness of the jargon used, and the clarity of explanations. The **completeness** of the documentation can be assessed by looking to see if it explains in sufficient detail how to carry out translation itself, dictionary editing, translation memory manipulation, and any other tasks. The documentation might also mention explicitly any limitations of the system.

3. Using translation software with language learners

A small number of writers have addressed the relationship between MT and language teaching and translator training.⁷ It has sometimes been suggested also that translation software can be used in the teaching of foreign languages. Obviously, inasmuch as translation is often part of foreign-language learning, we can say that learning about MT and translation software should be part of the curriculum for language learners. But some researchers have gone further and suggested that

translation software can be used to reinforce various aspects of the language-learning task. In this respect, the suggestion is that translation software can be used as a CALL (computer-assisted language learning) tool. A pioneer in this field was Corness (1985), who reported using the now defunct MT system *Alps* as a learning aid. Early releases of Globalink's software (e.g. *French Assistant*) promoted its use also as a language-learning aid, and the software included grammar tutorials (for example, lists of inflection paradigms). The field of CALL has developed independently over the years, and there are a great number of specific computer-based tools available for language teaching. The quality, complexity and sophistication of these tools vary enormously.

In the sections below we will concentrate on the use of MT "proper"⁸ by language learners. But many of the translation tools that we have discussed in this book can also serve a purpose for language learners. On-line dictionaries and thesauri have an obvious place in computer-based language learning. Corpus-based tools can be used very effectively with language learners, too. Translation memory and the related bilingual concordance both serve as a supplement to traditional dictionaries which, by their nature, can only contain limited examples of language in context. DeCesaris comments that

This type of program can be adapted for use in a classroom setting, because translation memories can be used as a self-learning resource to provide students with immediate access to models that they know are correct.

(DeCesaris, 1995: 264).

3.1 Producing a commented translation

Translation software is generally not designed with language learners in mind. For this reason, one should be a little wary of using a tool for a purpose that it was not originally intended. As already mentioned, **translation** is an exercise that features widely in language learning curricula, and so language students should be aware of translation software. As Lewis puts it,

...language graduates need to know what the capabilities of state-of-the-art MT are and how to evaluate its role as a practical tool in the language industry.

(Lewis 1997: 255f.)

[F]uture employers may expect prospective graduates in modern languages to have sufficient skills and background knowledge in translation technology to influence decisions on whether or not to invest in MT. (*ibid.*, page 261)

We have found it useful with fairly advanced students to ask them to use software to produce a first draft translation (into their native language) and then to produce an improved version (post-edited), together with a **commentary**. Where they have

had some classes about the general difficulties and problems of MT, we ask them to relate errors in the text to problems we have discussed in class. Alternatively, we can ask them to try in their commentary to classify on a linguistic or pragmatic basis the kinds of mistakes the translation software has made.

3.2 Using MT as a bad model

Another, more controversial, use of translation software in language learning is to use its weaknesses and mistakes to bring out subtle aspects of language differences or to reinforce learners' appreciation of both L1 and L2 grammar and style. Anderson (1995) describes use of a bidirectional English–Hebrew MT system in this way. Students manually entered sentences one by one from a suitable text corpus provided to them, noted the results, and then use **native-speaker intuition** and/or **L2 reference works** (depending on the translation direction) to identify and correct the errors. For translation into the L1, this can be a useful exercise, since the poorer-quality translations are usually too close to the lexical and syntactic structure of the source language, and this exercise can reinforce the students' awareness of differences between the languages by showing them a bad translation into their own language. Of course, a generally low-quality translation is not of interest *per se*; rather, the text should be used (and the original source text chosen so as to bring this out) to focus on particular phrases and constructions. Lewis endorses this approach too.

[M]any students have expressed the view that they have increased their cognitive knowledge of German grammar through having to enter information in the system's dictionaries; for those students whose command of formal grammar is weak, the MT dictionaries appear to provide a stimulus for researching areas of grammatical structure. (Lewis 1997: 270)

On the other hand, using this technique with translations into the second language carries with it the danger of reinforcing or even introducing incorrect language habits on the part of the learner. Students have a natural “respect” for the printed word, and there is a tendency for them to believe that the system is an authority on the target language, and so anything that it produces must be correct. This is of course a misapprehension of which they must be disabused.

Richmond (1994) overcomes this problem by providing a model translation. His use of the translation software to bring language contrasts to the attention of students is somewhat idiosyncratic, but may prove to be an enjoyable exercise which “makes a change” for some students. Richmond provides sample texts for translation into the second language — French in his case — along with model answers. Students are asked to type in the original (English) sentence, and note that

the system gets the translation wrong. They are then asked to try to modify the English sentence and retranslate it, continuing to do so until the appropriate target text is obtained. The idiosyncratic aspect of this however is that, because the translation software he uses tends to produce rather literal translations, in order to get the desired output, the original English text has to be modified *to make it more like the French target text!* As he points out,

This is, of course, the reverse of normal student behaviour, which so often consists of producing incorrect French that sounds like English. (Richmond 1994: 71)

He calls this “doing it backwards”, and the pedagogic reasoning behind this is that it causes the student to focus on the differences between French and English, and to “recognize the processes by which a given meaning is expressed in French” (*ibid.*, p. 72). Richmond goes on to state

...by increasing the students’ awareness of the differen[c]es between their first language and the target-language, the backwards translation method places the emphasis on linguistic *processes* and linguistic input rather than on linguistic forms and output. (p. 74; emphasis original)

The method has the advantage that “there is no danger that they will reinforce their own target-language errors” (p. 75), and a further pedagogical aspect of the exercise is that the “strange and often humorous” L1 constructions produced by the students help to fix the correct L2 constructions in their minds. Ball (1989) has a similar approach in which the student is invited to reconstruct the (English) source text from a raw translation such as (1a) of the French (1b).

- (1) a. He is older than I am not it.
b. *Il est plus âgé que je ne le suis.*

There may be something in this: surely all language learners at some point amuse themselves and their colleagues by imposing L2 constructions on their native language for comic effect? (Long-term expatriates introduce into their native language vocabulary and some turns of phrase from the language of the “host country” either deliberately, for ease of expression amongst their co-expats, where the local language has a “neater” way of saying something, or they may even do this subconsciously as their “idiolect”⁹ absorbs linguistic titbits.) Anecdotal evidence from Richmond is that students enjoy the exercise and find working with the translation software challenging and worthwhile. Perhaps just from its novelty value the exercise may be worth trying.

3.3 *TransIt TIGER*

The TELL consortium (Technology Enhanced Language Learning) of the Centre for Modern Languages, University of Hull, has produced a PC-based program, *TransIt-TIGER*, specifically aimed at translator training, as part of its contribution to the Computers in Teaching Initiative, a major (UK) national project to create CALL software for higher education. *TransIt-TIGER* is not a translation tool, but a program designed to help language-learners accomplish the task of translation, drawing their attention to and giving them help with various aspects of this task. Originally devised for English–Italian translation, the software was later generalised into an “authoring” package which enables teachers to develop material for any language pair.¹⁰

The approach is based on a two-stage activity. First, learners are given as much help as possible to enable them to produce their first effort at translating a passage. The kinds of help available include specialist glossaries, a dictionary, a thesaurus, and pre-prepared questions or hints. These hints focus on linguistic or grammatical points which are likely to cause difficulty, and generally ask questions which are intended to direct the learner towards an appropriate solution. Figure 2 shows the system in its “Hints” mode. As can be seen in the figure, the learner can also access a text-oriented glossary.

Students work with the system to produce a first draft translation which is then assessed by the teacher who draws attention to any areas which require further attention. In the second stage the learners are provided with two alternative translations. These are not model translations as such, but function as stimuli. The students now polish their first version in the light of the versions provided, to arrive at a final translation.

4. Teaching MT to end-users

4.1 Business people, scientists and technical writers

A category of “students” of MT which receives very little attention is the end-user. Typically, many types of end-user are not to be found in a student environment in the first place, which might explain the lack of attention to their needs. Some are however available for some sort of training, among them for example people who might want to use MT for assimilation or dissemination as part of their job. Miyazawa (2002) for example describes a scheme for training Japanese business people who need to read English on a day-to-day basis, and who are happy to use generally good quality commercially available Japanese–English MT systems to get

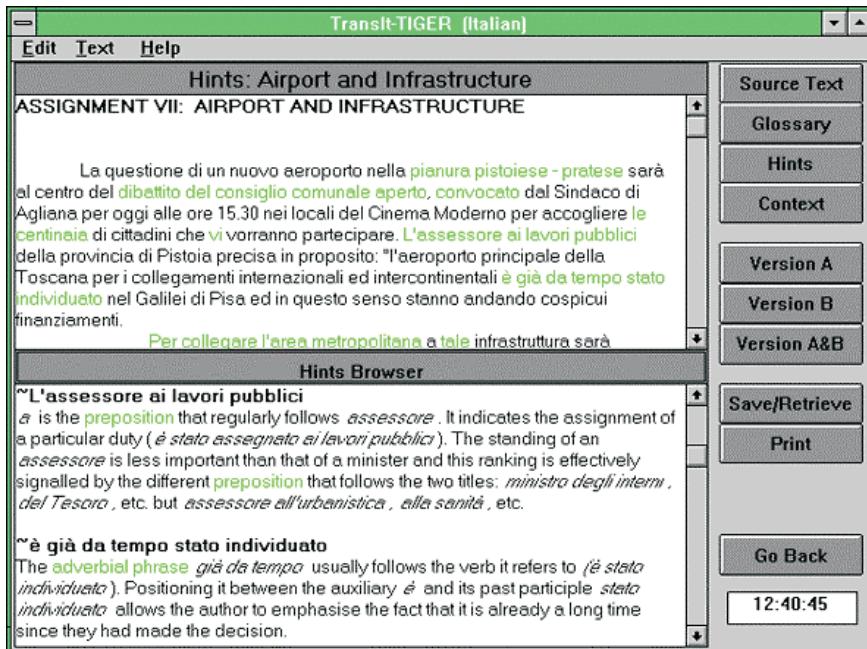


Figure 2. *TransIt-TIGER* in “Hints” mode¹¹

gist translations of documents, often with a short shelf-life, such as agendas and minutes of meetings. With a small amount of training and education concerning the problems of MT, they can gain more benefit from the MT systems through their knowledge of their limitations.

Similarly, end-users using MT for dissemination can be encouraged to exercise controls over their writing in order to ensure better quality output, as discussed in Chapter 14. Certainly, technical writers could benefit from a brief exposure to the topic, too.

4.2 Assessing MT for assimilation

A typical problem in teaching MT to students, whatever their background, is “language-pair deficiency”: if the class is quite heterogeneous, it may be that none of the systems that you happen to have in your “lab” cover language pairs that are suitable for all your students. In particular, you may have students who want to work in languages for which there are as yet no commercially available systems. In this case, a good assignment is to focus on the “for assimilation” function of translation software, where it is used to produce a rough gist of an otherwise

Figure 3. Example of Russian web page

unreadable text. Ask students to find a text (on the World Wide Web for example) in a language which is covered by the systems at your disposal but which is unfamiliar for them. For example, Figure 3 shows a Russian web page;¹² the translation obtained from AltaVista's *babelfish* is shown in Figure 4. The translation is not excellent, but it is certainly understandable for the most part. An additional exercise might be to see whether it is possible to post-edit the text in Figure 4 up to some imagined “publishable” standard, without seeing the source text and without any knowledge of the source language (obviously this exercise is different for someone who *does* know Russian).

5. MT and related disciplines

Historically, MT was probably the first proposed non-numerical use of computers. From early (not entirely successful) attempts to use computers to translate natural languages grew the now well-established field of Computational Linguistics (CL). This field can be characterised as the use of the computer in any activity involving language (both written and spoken), and like many other fields has its theoretical, methodological and practical sides. As a branch of linguistics it also has its theoretical “schools” as well as a certain body of accepted lore and practice. Related to CL is the field of Linguistic Computing, which can be said to be the use of computers in

General information

History of university, regulations, structure, information about the management/manual, the departments, the public organizations of university.

For those entering

Rules of method for the high school seniors, into the graduate study, into the doctoral students/doctoral study, for obtaining second higher education. Addresses and the telephones of acceptance boards. Information about the dovuzovskoy preparation.

Training process

Training programs, the program of training courses, lecture materials. Remote instruction. Release/issue. Poslevuzovskoye formation.

Scientific work

Of theses, Lomonosovskiys are reading, the priority directions of scientific studies, Grants, etc.

Publications Of MGU

the “ herald of Moscow University “, the newspaper “ Moscow University “, the scientific library OF MGU, other scientific training publications OF MGU.

To students

Student servers and page, student publications, other resources/lifetimes for the students.

Resources/lifetimes It internet

The list of the web- sites OF MGU, network/grid MSUNet, Russian scientifically-educational resources/lifetimes.

Addresses

Figure 4. *Babelfish*'s translation of text in Figure 3.

relation to more traditional and established areas of research in linguistics. A good example is literary studies, particularly “stylistics” (for example, authorship studies — did Shakespeare write all the plays usually attributed to him?) which has been revolutionised by the use of computers to store and analyse text. The computer's capacity for storage and swift (numerical) analysis impinges on other branches of linguistic science, particularly viewed as a social science.

Coming back to CL as a branch of linguistics in itself, we can identify, as in “general” linguistics, basic theoretical and methodological aspects applied to the various “strata” of language description that linguistics generally recognises: phonetics and phonology (speech sounds), orthography (writing and spelling), morphology (the internal structure of words, including inflections etc.) and word-formation, syntax and grammar, semantics and meaning, pragmatics and usage. CL focuses on computational aspects of the above, notably **representation** (in a computationally tractable manner), **analysis** (i.e. mapping from one, more superficial, level of representation to another more abstract), and **generation**, the inverse. In CL we can also recognise numerous **applications** of these fundamental method-

ologies, translation being one, along with speech-to-text conversion and vice versa, text summarization, information extraction, language-based human-computer interaction, and other computer-mediated uses of language.

The most interesting aspect of MT for CL is that, more than any other application, translation requires “coverage” of all the linguistic levels in more than one language. For this reason MT is sometimes seen as the archetypical application of CL. Another useful feature of translation as a test-bed for CL techniques is that you can usually tell pretty well whether an MT program has “worked” (notwithstanding subtle difficulties of saying just how “good” a translation is, it is usually quite clear whether some piece of text is or is not a translation of another text).

For the student (and teacher) of CL, then, translation software can be used to illustrate problems (and solutions) in language analysis at various levels both monolingually and contrastively. Source-text analysis requires morphological disambiguation (is a *tower* a high structure or something that tows?) and interpretation (is *books* the plural of *book*, or a form of the verb *to book*?), word-sense disambiguation (*bank*: financial institution or side of a river?), syntactic, semantic and pragmatic disambiguation. Translation involves converting linguistic aspects of the source text into their appropriate form in the target text, thus the application of contrastive lexical and syntactic knowledge. And the generation of the target text involves the corresponding problems of style, syntax, and morphology.

Exercises can be developed to familiarize students with weaknesses and problems of translation software (these can also be used for trainee translators). A suite of the “trick” sentences like the ones illustrated in the Section 2 will be suitable.

More generally, translation software output can be used with students of CL for linguistic error analysis in general or focussing on one particular problem area, using a specially designed test suite. For example, if one was interested in the subtleties of modality (in English, expressed by words like *can*, *must*, *should*, *ought to*, etc.) one could construct a set of sentences which express different modalities, and see how they are translated. Other interesting linguistic phenomena which illuminate contrastive differences between languages are the use of tenses, (in)definiteness, passive constructions and other means of topicalisation, and so on. Lewis (1997) shows an example of a test suite of sentences for use with translation software to investigate the translation of complex English verb forms into German. Finally, a test suite can be used to explore the linguistic rules apparently used by the system (“reverse engineering”). Some of the examples above adopt this approach: for instance, the first German case explores whether the idiomatic translation *Hunger haben* → *be hungry* is maintained when the phrase is modified by an adjective in German which is rendered as an adverb in English.

More peripheral to our interests, MT offers some interesting computational problems for computer scientists, though looking at commercial software is not an

especially productive way of investigating these, since it is difficult to get much information on how most commercial translation systems really work. A number of centres aim to teach students how to *develop* MT systems since, after all, commercial software has to be written by someone, and such people need appropriate training. Kenny and Way (2001) and Somers (2001), among others, have emphasised the distinction between students of CL and of translation studies. Consecutive papers at the 2002 “Teaching Machine Translation” workshop¹³ dealt with MT from a more computational standpoint.

Further reading

There is a small but growing literature on the topic of teaching MT. In particular, at the time of writing there have been two dedicated workshops, at the 2001 MT Summit in Santiago de Compostela, and in 2002 at UMIST, Manchester. Kingscott (1996), Kenny (1999) and L'Homme (1999) describe their experiences of integrating translation tools into translator training curricula. Lynne Bowker's recent book (Bowker, 2002) starts with a useful chapter on why translators need to learn about technology and in general is to be highly recommended.

There are a number of MT textbooks that take a more technical approach than the one offered in this book, and discuss MT from a CL point of view. Amongst these are Hutchins and Somers (1992) and Trujillo (1999). General introductions to CL are too numerous to mention, but Allen (1995) is probably the most popular among teachers of CL, while Dale et al. (2000) will certainly give as much detail about particular aspects of CL as you could want.

For more information on CALL, Levy (1997) and Cameron (1999) both provide good overviews.

Notes

1. In the remainder of this chapter we will use the term “translation software” to mean any computer software relevant to the translation process, from computational tools for translators via computer-aided translation software through to fully automatic MT. And we will use the term “MT” in its most generic sense of “research in the development and use of translation software”. See my comments in Chapter 1 on the problems of terminology in this field.
2. Federal Association of Interpreters and Translators.
3. For example, Schubert (1993), Haller (1995).
4. See Reuther (1999). The LETRAC web page is at www.iai.uni-sb.de/letrac/home.html.

5. Screen shot from the *French Assistant* system.
6. Material in this section is based on Hartley and Schubert (1998).
7. For example, Loffler-Laurian (1983, 1985), Ball (1989), Mitkov et al. (1996), Lewis (1997).
8. That is, translation software that performs some or all of the translation task automatically.
9. This is the term linguists use to indicate the form of language used by a single individual: their idiosyncratic dialect.
10. Thompson et al. (1996). The following description of *TransIt* is based on various sources, notably Thompson (1996), Burnage (1998), and web pages, especially www.ilt.ac.uk/resources/publications/al_archive/issue4/thompson2.htm.
11. Screenshot taken from Thompson (1996).
12. This is part of Moscow State University's home page, www.msu.ru, © Moscow State University.
13. Way (2002), Amores (2002), v. Hahn and Vertan (2002), Sheremetyeva (2002).

References

- Allen, James F. (1995) *Natural Language Understanding*, 2nd edition. Menlo Park, CA: Benjamin Cummings.
- Amores, J. Gabriel (2002) "Teaching MT with XEpisteme", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 63–68.
- Anderson, Don D. (1995) "Machine Translation as a Tool in Second Language Learning", *CALICO Journal* 13.1, 68–97.
- Ball, R. V. (1989) "Computer-assisted Translation and the Modern Languages Curriculum", *The CTISS File* 8, 52–55.
- Bowker, Lynne (2002) *Computer-Aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.
- Bundesverband der Dolmetscher und Übersetzer (BDÜ) (1986) "Koordinierungs-ausschuß Praxis und Lehre", *Mitteilungsblatt für Dolmetscher und Übersetzer* 32.5, 1–8.
- Burnage, Gavin (1998) "Teachers and Technicians: Working Together for Effective Use of Information Technology", in Sarah Porter and Stuart Sutherland (eds) *Teaching European Literature and Culture with Communication and Information Technologies: Selected Papers*, Oxford: CTI, Oxford University Computing Laboratory. Available online at www.mml.cam.ac.uk/itmml98.html.
- Cameron, Keith (ed.) (1999) *Computer-Assisted Language Learning (CALL): Media, Design and Applications*. Lisse: Swets & Zeitlinger.
- Corness, Patrick (1985) "The ALPS Computer-assisted Translation System in an Academic Environment", in Catriona Picken (ed.) *Translating and the Computer* 7, London: Aslib, pages 118–127.

- Dale, Robert, Hermann Moisl and Harold Somers (eds) *Handbook of Natural Language Processing*. New York: Dekker.
- DeCesaris, Janet Ann (1995) "Computerized Translation Managers as Teaching Aids", in Cay Dollerup and Vibeke Appel (eds) *Teaching Translation and Interpreting 3: New Horizons*, Amsterdam: John Benjamins, pages 263–269.
- Haller, Johann (1995) "Computerlinguistik und maschinelle Übersetzung in einem Studiengang für Übersetzer und Dolmetscher", *LDV-Forum* 12, 29–34.
- Hartley, Tony and Klaus Schubert (1998) "From Testbench to Workflow: Relocating MT in Education and Training", in *Translating and the Computer 20: Proceedings of the Twentieth International Conference on Translating and the Computer*, London. [no page numbers]
- Hutchins, W. John and Harold L. Somers (1992) *An Introduction to Machine Translation*. London: Academic Press.
- Kenny, Dorothy (1999) "CAT Tools in an Academic Environment", *Target* 11, 65–82.
- Kenny, Dorothy and Andy Way (2001) "Teaching Machine Translation & Translation Technology: A Contrastive Study", in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, Spain, pages 13–17.
- Kingscott, Geoffrey (1996) "The Impact of Technology and the Implications for Teaching", in Cay Dollerup and Vibeke Appel (eds) *Teaching Translation and Interpreting 3: New Horizons*, Amsterdam: John Benjamins, pages 295–300.
- Levy, Michael (1997) *Computer-Assisted Language Learning: Context and Conceptualization*. Oxford: Clarendon Paperbacks.
- Lewis, Derek (1997) "Machine Translation in a Modern Languages Curriculum", *Computer Assisted Language Learning* 10, 255–271.
- L'Homme, Marie-Claude (1999) *Initiation à la traductique*. Brossard, Québec: Linguatech.
- Loffler-Laurian, Anne-Marie (1983) "Traduction automatique et enseignement", *Revue de Phonétique Appliquée* 66–68, 86–102.
- Loffler-Laurian, Anne-Marie (1985) "Informatique, traduction et enseignement des langues", *Meta* 30, 274–279.
- Mitkov, R., J. Higgins-Cezza and O. Fukutomi (1996) "Towards a More-efficient [sic] Use of PC-based Machine Translation in Education", in *Translating and the Computer 18: Papers from the Aslib conference*, London. [no page numbers]
- Miyazawa, Orie (2002) "MT Training for Business People and Translators", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 7–12.
- Mowatt, David and Harold Somers (2000) "Is MT Software Documentation Appropriate for MT Users?", in John S. White (ed.) *Envisioning Machine Translation in the Information Future: 4th Conference of the Association for Machine Translation in the Americas, AMTA 2000*, Berlin: Springer, pages 223–238.
- O'Brien, Sharon (2002) "Teaching Post-editing: A Proposal for Course Content", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 99–106.
- Pérez-Ortiz, Juan Antonio and Mikel Forcada (2001) "Discovering Machine Translation Strategies Beyond Word-for-Word Translation: A Laboratory Assignment", in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, Spain, pages 57–60.

- Reuther, Ursula (ed.): Toni Badia, Carme Colominas, Mary Filippakopoulou, Karl-Heinz Freigang, Dagmar Fuchs, Johann Haller, Christoph Horschmann, Peter Kastberg, Marie Kosmarikou, Belinda Maia, Bernt Moeller, Jennifer Pearson, Paul Schmidt and Maria Tsoutsoura (1999) "LETRAC Survey Findings in the Educational Context", Deliverable D1.2, European Commission DG XIII Telematics Application Programme Fourth Framework, project LE4-8324, available at www.iai.uni-sb.de/letrac/D12.doc.
- Richmond, Ian M. (1994) "Doing it backwards: Using translation software to teach target-language grammaticality", *Computer Assisted Language Learning* 7, 65–78.
- Schubert, Klaus (1993) "Zwischen Benutzerschulung und Wissenschaft: Sprachtechnologie in der Übersetzerausbildung", in Horst P. Pütz and Johann Haller (eds), *Sprachtechnologie: Methoden, Werkzeuge, Perspektiven*, Hildesheim: Olms, pages 304–311.
- Sheremetyeva, Svetlana (2002) "An MT Learning Environment for Computational linguistics Students", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 79–87.
- Somers, Harold (2001) "Three Perspectives on MT in the Classroom", in *MT Summit VIII Workshop on Teaching Machine Translation*, Santiago de Compostela, Spain, pages 25–29.
- Thompson, A. D., J. Thompson and P. Corness (1996) *TransIT-TIGER: Authoring Shell*. London: Hodder & Stoughton Educational.
- Thompson, June (1996) "Embedding Technology into Language Examinations", *Active Learning* 4, online journal published by the Institute for Learning and Teaching in Higher Education, www.ilt.ac.uk/resources/publications/al_archive/issue4/thompson/default.htm.
- Trujillo, Arturo (1999) *Translation Engines: Techniques for Machine Translation*. London: Springer Verlag.
- v. Hahn, Walther and Christina Vertan (2002) "Architectures of 'Toy' Systems for Teaching Machine Translation", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 69–77.
- Way, Andy (2002) "Testing Students' Understanding of Complex Transfer", in *6th EAMT Workshop Teaching Machine Translation*, Manchester, pages 53–61.

Key to exercises

French examples:

1. *MT output*: The bird entered the room. The bird entered the room hopping.

Preferred translation: The bird flew into the room. The bird hopped into the room.

French verbs of motion incorporate the direction (*entrer* ‘enter’, *traverser* ‘cross’) and must specify the manner separately (*en courant* ‘running’, *à la nage* ‘swimming’). In English, it is the other way round: *run into*, *swim across*. In the second example, the best translation is *hopped into*. This gives a clue also to the translation of the first example: *enter* is of course the literal translation, but *flew into* might be more natural if you consider that French would not say *entra en volant* if the subject is a bird.

2. *MT output*: Charles suicided himself.

Preferred translation: Charles committed suicide.

Does the dictionary have this semi-idiomatic phrase?

3. *MT output*: One gave the book to Paul. One has slept in this bed.

Preferred translation: Paul was given the book. This bed has been slept in.

French uses the impersonal construction with *on* whereas in English a passive construction is more natural: in French, passives can be formed by “promoting” direct objects to subject position, but not indirect objects or prepositional objects.

4. *MT output*: We come from finish to read this book.

Preferred translation: We have just finished reading this book

Another idiomatic construction.

5. *MT output*: My cousin is beautiful. My cousin is beautiful. My cousin is rich.

Preferred translation: My cousin is beautiful. My cousin is handsome. My cousin is a rich woman.

This example involves “compensation”: French distinguishes the gender of *cousin(e)* whereas English does not. On the other hand, English distinguishes *beautiful* and *handsome*, so we can compensate for the neutral gender of *cousin* by choosing an adjective which shows the sex of the cousin. A different strategy must be adopted for *rich* however.

6. *MT output*: The feet of the table are very thick.

Preferred translation: The legs of the table are very thick.

A matter of lexical choice: tables have *pieds* ‘feet’ in French, but *legs* in English.

7. *MT output*: I hired the car from Avis. Avis hired me the car.

Preferred translation: I loaned the car from Avis. Avis rented me the car.

Again, a matter of lexical choice: *louer* in French covers the hiring transaction in both directions, whereas English lexicalises the role of the subject. Some MT systems might avoid the problem by translating *louer* as *hire*, which has the same ambiguity.

Another (unrelated) problem is that the proper name *Avis* might be translated as *Opinion* in one or both of the examples.

8. *MT output*: The thief gave a kick to the policeman. The thief gave violent hits of the foot and the fist to the policeman.

Preferred translation: The thief kicked the policeman. The thief violently kicked and punched

the policeman.

The problem here is that French has no single lexical equivalent to *kick* and *punch*. On its own, *donner un coup de pied à* may be correctly (or partially correctly) translated, but in the second example, the translation is complicated by the adjective which should be translated as an adverb, and the conjunction with the parallel construction *donner un coup de poing à* ('punch').

9. *Preferred translation:* The pilot closes the door. The agile pilot carries it.

The second example is designed to draw your attention to an alternative interpretation of the first, namely 'The firm pilot carries her.'

10. *MT output:* You can do your shopping from your domicile.

Preferred translation: You can do your shopping from your home.

The more literal translation is stylistically inappropriate.

11. *MT output:* My ancient husband visited an ancient ruin.

Preferred translation: My former husband visited an ancient ruin.

The translation of *ancien* depends on its position relative to the noun it modifies.

German examples:

1. *MT output:* I am hungry. I have big hunger.

Preferred translation: I am hungry. I am very hungry.

The system has the straightforward idiom in its dictionary, but is unable to handle the modified version of it.

2. *MT output:* I eat gladly. Fritz often plays tennis gladly.

Preferred translation: I like eating. Fritz likes to play tennis often.

This is an example of a "head-switching" translation: the "head" in the German is the verb *essen* 'eat', while in English it switches to *like*. Even if the system gets the simple case correct, the difficulty in the second example is knowing where to attach the adverb: would *Fritz often likes to play tennis* be acceptable?

3. *MT output:* The girl pleases the man. The girl seems to please the man.

Preferred translation: The man likes the girl. The girl seems to be liked by the man.

This time we have a case of "structural change", because the syntactic roles of subject and object have to be reversed if we translate *gefallen* as *like* rather than *please*. The complication arises in the second example where *Mädchen* functions as the subject of both *scheinen* 'seem' and *gefallen*. To preserve this in English we either have to keep the more literal translation or find a construction which keeps *girl* as the subject.

4. *MT output:* It is danced and eaten.

Preferred translation: There is dancing and eating.

This particular use of the passive, especially with intransitive verbs, should not be translated literally.

5. *MT output:* Hans wants, that Kurt eats its breakfast.

Preferred translation: Hans wants Kurt to eat his breakfast.

The infinitive construction is preferable to the more literal translation. Also, the MT system may not get the correct possessive pronoun reference.

6. *MT output*: I love cross word puzzles. Hans is a fast Kreuzworträtsellöser.

Preferred translation: I love crossword puzzles. Hans is a fast crossword puzzle solver.

Systems translating from German must be able to interpret novel compounds which are not in the dictionary. Some systems can make a reasonable attempt at simple ones, but may founder in the face of more complex ones. To change the second example round into something more natural like *Hans is quick at solving crossword puzzles* is an even bigger step for MT.

7. *MT output*: Shoplifter steel is here an important problem.

Preferred translation: Shoplifting is a big problem here.

Another example of an “unknown” compound which the system incorrectly disentangles.

8. *MT output*: My wristwatch proceeds.

Preferred translation: My wristwatch is fast.

The translation *be fast* for *vorgehen* is a particularly special case. Some systems may also have trouble with *Armbanduhr* if it is not in the dictionary, since its components are individually highly ambiguous — I know of one system which produced *poor volume time*.

9. *MT output*: The former chancellor is called cabbage. Mr Kohl is now retired.

Preferred translation: The former chancellor is called Kohl. Mr Kohl is now retired.

This is the classic problem of translating proper names. Many systems recognise that the word following *Herr ‘Mr’* is probably a name, but identifying names which are also common words is more difficult in isolation.

10. *MT output*: The doves leave the buildings in the city centre very dirty. The dove brought back the olive branch.

Preferred translation: The pigeons leave the buildings in the city centre very dirty. The dove brought back the olive branch.

The German word *Taube* has two translations in English, *pigeon* or *dove*. Real-world knowledge, including knowledge of folklore, determines the choice in these two examples. This is impossible for an MT system to do systematically.

11. *MT output*: My cousin is beautiful. My cousin is beautiful. My cousin is rich.

Preferred translation: My cousin is beautiful. My cousin is handsome. My cousin is a rich woman.

This example involves “compensation”: German distinguishes male and female cousins as *Vetter* and *Kusine* whereas English does not. On the other hand, English distinguishes *beautiful* and *handsome*, so we can compensate for the neutral gender of *cousin* by choosing an adjective which shows the sex of the cousin. A different strategy must be adopted for *rich* however.

12. *MT output*: In this university study 3 000 students and students.

Preferred translation: There are 3,000 students in this university.

The main point of this example is a kind of corollary to the previous case: since English does not distinguish male and female students, a single translation is sufficient. Another problem is that the literal translation of the verb *studieren* sounds unwieldy. Finally, does your MT system correctly punctuate numbers?

Index

A

- ABLE International 298, 299, 306
ACE (Attempto Controlled English) 251
Adams, D. 209
adequacy 236–8
Adriaens, G. 277, 278, 279
AECMA 246, 249, 250, 251, 254, 278, 279
Ahmad, K. 29, 63
alignment 34–7, 46
Albanian 90
Allegranza, V. 158, 159
Allen, J. 299, 313, 315, 316
Allen, J.F. 334, 335
ALPAC report 4–5, 13, 14, 212, 228–9, 238, 242
Alps 33, 169, 326
Al-Shabab, O.S. 107, 115
Alta Vista 171, 191–206, 209, 300, 321, 331
ambiguity 124, 127, 131, 137, 139, 141, 247, 260, 265, 273, 285, 287, 292
 lexical 124, 150, 310, 333
 structural 124
Ament, K. 209
Amikai 206–8
Amores, J.G. 335
analysis 122, 145, 332–3
 problem 123, 124–8
anaphora 125, 157
Anderson, D.D. 327, 335
approved 246, 252
Arabic 90, 92, 93, 107, 172, 180
Arad, I. 101
architecture 122–3, 146, 163
Arnold, D.[J.] 10, 11, 141, 158, 159, 242, 243
ARPA 213, 243

Arthurn, P. 32–3, 46

- Ashworth, D. 29, 30
assimilation 6, 162, 180, 201, 301, 330–1
Association for Machine Translation 11, 173, 299, 312
attributes 216, 223
Austermühl, F. 63, 64
autocorrection 93

B

- babelfish* 171, 191–206, 300, 321, 331
Baker, K.L. 273, 279
Baker, M. 109, 113, 114, 115, 116, 147, 159
Balkan, L. 11, 141, 279, 280
Ball, R.V. 328, 335
Bar-Hillel, Y. 4, 5, 189, 190
Basic English 250
BDÜ (*Bundesverband der Dolmetscher und Übersetzer*) 320, 335
Belsie, L. 209
Bengali 90, 173
Benis, M. 16, 29, 30, 42, 46
Bennett, P. 158, 159
Bennett, W.S. 208, 209, 315, 316
Biber, D. 294, 295
black-box 217, 225–6, 236
Bourigault, D. 64
Bowker, L. 29, 45, 113, 115, 116, 334, 335
Brent, M.R. 100, 102
Bressan, D. 158, 159
Brigham Young University 33
Brighton University 111
BSEC 254, 278
Budin, G. 63, 65
Burnage, G. 335

- C
 Cabré, M.T. 63, 64
 CALL (computer-aided language learning) 326, 329, 334
 Callison-Burch, C. 208, 209
 Cameron, K. 334, 335
 Campbell, D.T. 242, 243
 Canadian 26, 28, 111, 289
 Cantonese 90, 91–2
 Cap Gemini 256, 278
 Carbonell, J.G. 279, 280
 Carnegie Group Inc. 250, 254, 279
 Carnegie Mellon University 250, 259
 Carroll, J.B. 228
 CASE (Clear and Simple English) 250
 Case, J.I. 250
 CASL 250, 307
 CAT, *see* computer-aided translation
 Caterpillar Inc. 250, 254, 259, 298, 299, 304
see also CFE, CTE
 Celtic languages 172
 CFE (Caterpillar Fundamental English) 250, 261
 Chandioux, J. 289, 293, 294
 Charrow, V.R. 294, 295
 chat-rooms 170, 205, 206–8
 Chervak, S. 258, 279, 316
 Chinese 90, 91–2, 131, 163, 172–3, 180
 Choi, S-K. 208, 209
 Church, K.W. 29, 30, 99, 102
 CL, *see* controlled language
 Clark, R. 58, 64
 CLAW 278, 279, 312
ClearCheck 254, 270
 combinatorial explosion 122, 126, 127, 132, 139
 commercial MT systems 21–5, 161–74, 175–90
 Commission for Racial Equality 88, 102
 common-sense 121, 126, 182
 communication 162, 181, 201
Compendium of Translation Software 164, 173
 CompuServe 197, 198
 computer 120–2
 -aided translation (CAT) 6, 13, 93
 science 6, 333–4
 Comrie, B. 158, 159
 concordance 25–8, 98, 107, 111, 112
 Condamines, A. 63, 64
 consistency 248, 256, 262
 contrastive analysis 145, 150, 223
 controlled language (CL) 245–81, 283, 288, 298, 322, 323
 checker 251–3, 275, 278
 correction 253–4
 Cormier, M.C. 29, 30
 Corness, P. 326, 335
 corpus 8, 25–7, 33, 60, 98, 99, 105–17
 comparable 106, 112, 114
 English (ECC) 109
 parallel 8, 25, 33, 99, 106, 108–12, 139–40, 179
 Canadian Hansard 26, 28, 139
 English Norwegian (ENPC) 109, 110
 Translational English (TEC) 113
 cost 42, 231, 232
 coverage 187, 196, 223, 225, 283
 Cremers, L. 190
 Croatian 90
 CTE (Caterpillar Technical English) 250, 259–76, 278
- D
 Dale, R. 334, 336
 Dagan, I. 99, 102
 Danish 94, 112, 150
 Darbelnet, J. 152, 158, 160
 DARPA 221, 235–9
 database 33–4
 data-driven 176, 178–9, 188
 Dauphin, É. 278, 280
 debugging 185, 206
 DeCesaris, J.A. 326, 336
 Defrise, C. 159
Déjà Vu 32, 41, 80
 de Koning, M. 278, 279, 280, 281
 description, level of 144–5, 332

- problem of 127–9
 desk-top publishing 18, 78, 90
 dialog box 71, 78
 dictation tools 16–17
 dictionaries 7, 19, 90, 96, 98, 183, 323
 direct approach 122–3
 dissemination 6, 161, 180, 201, 275, 301, 303, 330
 documentation 42, 73, 232, 325
 domain 129, 165, 172, 182
 Dorr, B.J. 152–3, 158, 159, 243
 Dostert, B. 242, 243
 draft translation 19, 22
 Drury, C. 279, 316
 Dubuc, R. 58, 64
 Dunlap, B. 299, 316
 Dutch 129
- E**
 EAGLES 46, 106, 116, 228, 242, 243
 Ebeling, J. 110, 116
 EBMT, *see* Example-based MT
 EC, *see* European Commission
 Eco, U. 202
 e-mail 7, 18, 162, 170, 187, 192, 303, 304, 324
 English 5, 6, 9, 15, 18, 35, 58, 59, 79, 87, 88, 93, 107, 109–11, 129, 133, 149, 154, 156, 164, 172, 180, 191, 196, 206, 245–81
passim, 312, 313
 - American 8, 11, 87
 - British 11
 - Canadian 11, 111
 - Chinese 99
 - French 6, 53, 111, 120, 129, 139, 141, 164, 191, 198, 202, 203–4, 207, 283, 289–95, 313, 327–8
 - German 129, 130, 164, 191, 198, 333
 - Hebrew 327
 - Italian 112, 151, 164, 191, 329
 - Japanese 26, 44, 131, 164, 169, 170, 172
 - keyboard 87
 - Norwegian 109
- Portuguese 109, 146, 191, 201
 -Spanish 152, 164, 168, 191, 202, 203–4, 216, 225–6
 translational 109, 113–4
 -Urdu 99
- entertainment 201–2
 equivalence 119–20
 Esselink, B. 85, 86
 Estival, D. 29, 30
Eurodicautom 14, 50, 311
 European Commission (EC) 5, 14, 85, 169, 192, 298, 299, 302, 320
 - Translation Service (ECTS) 311–12
 evaluation 41–3, 46, 63, 199–200, 211–44, 257–9, 276, 322
 Example-based MT (EBMT) 44–5, 100, 136–7, 178
 explication 109, 113
 extensibility 217, 223, 225, 232
- F**
 FAHQQT 6, 181
 Farsi 90
 Farwell, D. 101, 102, 238, 243
 Fawcett, P. 156, 158, 159
 feasibility 212, 222–3, 239
 fidelity 216–19, 227, 228–9, 236–7, 239
 Finnish 112
 Flanagan, M. 193, 197, 209, 227, 235, 243
 Flournoy, R.S. 208, 209
 fluency 236–8
 font 90, 195
 Forcada, M. 320, 336
 Fouvry, F. 279, 280
 Fox, B.A. 158, 159
 Frawley, W. 109, 116
 French 91, 93, 111, 112, 120, 125–6, 128, 130, 137, 149, 151–2, 164, 172, 180, 203–4, 206–8, 251, 289, 291–2, 309, 312, 313, 321, 327–8, 338–9
 - English 53, 112–3, 151–2, 191, 236–8, 313
 - keyboard 87*French Assistant* 22, 23, 24, 324, 326, 335

- Fuchs, N.E. 251, 280
 Fung, P. 99, 102
 fuzzy match 38–9, 43, 55, 80
- G**
 Gale, W. 99, 102
 Gaussier, E. 63, 64
 Gavioli, L. 112, 116
 Gawron, J.M. 11, 141
 General Motors 251, 298, 307
 Geoffroy-Skuce, A. 1 11, 116
 Gerber, L. 35, 178, 190, 208, 210, 242, 243, 316
 German 93, 95, 110, 112, 114, 129–30, 149–50, 1 53–5, 172, 180, 203–4, 206, 251, 258, 312, 321–2, 333, 339–40
 -English 35, 154, 164, 165, 172, 191
 -French 164
 -Italian 164
 gisting 162, 227, 300, 301–2, 304, 330
 Giussani, B. 209
 glass-box 217, 225–6
 globalisation 68, 299–300
 glossary 50, 56, 77
 Glover, A. 278, 280
 Godden, K. 305, 317
 Goyvaerts, P. 249, 280
 grammar 144
 checker 90, 96; *see also* CL checker
 graphical user interface 81
 Greek 90, 112, 172
 Gringas, B. 278, 280
 Grishman, R. 294, 295
 Gujerati 90
- H**
 Hajic, J. 100, 102
 Haller, J. 334, 336
 Halliday, M.A.K. 111, 115, 116, 156, 159
 Hamon, T. 63, 64
 Hansard, *see* corpus
 Harris, Z. 283, 295
 Hartley, T. 335, 336
 Hatim, B. 158, 159
- Havrila, R. 100, 102
 Hawkins, J.A. 150, 158, 159
 Hebrew 95, 172, 327
 Heid, U. 63, 64
 Heinisz-Dostert, B. 242, 243
 Helmreich, S. 238, 243
 Hervey, S. 158, 159
 Higgins, I. 158, 159
 Hindi 90, 173
 Hirschman, L. 242, 243
 Hirst, G. 158, 159
 history
 of localisation 83
 of MT 4, 212
 of terminology tools 50–1, 63
 of translation memory 32–3, 45
 of translator's workstation 13–14
 hit list 55–6
 Hoard, J. 278, 280, 281
 Hogan, C. 313, 316
 Holloway, T. 46
 Holmback, H. 258, 280, 281, 317
 Hornstein, N. 158, 159
 hot key 72, 78
 Hovy, E. 35, 243
 HTML 19, 68, 71, 73, 74, 195
 Humphreys, [R.] L. 11, 141, 159, 243, 278, 280
 Hungarian 114, 172
 Hutchins, [W.] J. 10, 11, 29, 30, 45, 46, 141, 158, 159, 173, 174, 242, 243, 294, 295, 334, 336
 hyphenation 93
- I**
 ideal translation 213, 223
 idioms 183, 199
 ill-formed input 124, 127
 inbound, *see* assimilation
 Indian languages 87, 173
 Indonesian 172
 informativeness 236
 intelligibility 216–19, 227, 229, 236, 239, 240

- interactive 23, 274
 interface structure (IS) 122, 146
 interlingua 123, 124, 131, 134, 152
 internationalisation 68
 Internet 19, 81, 83, 162, 163, 168, 170–1,
 172, 191–210, 300, 302
 intuition 215, 218, 219, 228, 241
 Inuktitut 293
 IS, *see* interface
 Isabelle, P. 29, 30, 99, 102, 294, 295
 Isahara, H. 242, 244
 Italian 112, 151, 164, 172, 180, 191, 204,
 206, 329
- J**
 Jackendoff, R. 153, 160
 Jacquemin, C. 63, 64
 Jaekel, G. 63, 64
 James, C. 145, 160
 Japanese 35–6, 44, 131, 163, 169–70, 172–3,
 180, 206, 277
 –English 6, 35, 164, 169–70, 172, 236–
 8, 329–30
 JEIDA 221, 232–5, 243
 Johansson, S. 109, 110, 116
 Johns, T. 112, 116
 Johnson, E. 278, 280, 281
 Jones, D. 100, 102
 Jordan, P.W. 242, 243, 280
 justification 92–3
- K**
 Kageura, K. 63, 64
 KANT 250, 256, 259–76, 277
 Kay, M. 10, 11, 13, 29, 30, 32, 46, 141
 Kehler, A. 159, 160
 Kelly, T. 209
 Kennedy, G. 102, 116
 Kenny, D. 59, 64, 113, 116, 334, 336
 keyboard 90–2,
 Kilby, K. 141, 142, 294, 295
 Kincaid, C. 258, 280
 King, P. 112, 113, 116
 Kingscott, G. 334, 336
- Kittredge, R. 281, 284, 285, 289, 294, 295
 Knight, K. 176, 190
 Koch, K. 63, 64
 Kohn, J. 114, 117
 Korean 163, 164, 172, 206, 207
 Krings, H. 315, 317
 Kugler, M. 29, 30
- L**
 Lambrecht, K. 158, 160
 Lange, C.A. 190
 Lange, E.[D.] 206, 208, 210, 242, 244
 LANT 254
 Laviosa[-Braithwaite], S. 108–9, 113, 117
 Lauriston, A. 63, 64
 Lehrberger, J. 284, 294, 295
 Lehrndorfer, A. 258, 278, 280
 León, M. 308
 LETRAC 85, 86, 320
 Levenshtein distance 39
 Levin, B. 158, 160
 Levy, M. 334, 336
 Lewis, D. 326–7, 333, 336
 lexical resources 19–22
 lexicon 135, 144, 155, 183; *see also*
 dictionary
 L'Homme, M.-C. 63, 64, 334, 336
 linguistics 143–60
 computational 2, 5, 14, 25, 92, 93, 144,
 331–4
 theory 144
 LISA 34, 45, 59, 67, 68, 84, 85, 86, 168
 Lispector, C. 108
 localisation 52, 67–86, 90, 167–8, 299
 definition 67
 history of 83
 industry, *see* LISA
 Löffler-Laurian, M.-C. 301, 315, 309–11,
 317, 335, 336
 Logos [company] 9, 190
 Logos [system] 164, 165, 177
 Lonsdale, A.B. 158, 160
 Luong, T.V. 86
 Lux, V. 278, 280

- M**
- Machine translation (MT) *passim*
second generation 5–6, 290
- Macintosh 163, 309
- Macken, L. 278, 279
- Mágan Muñoz, F. 101, 102
- Maia, B. 109, 117
- Malay 172
- Malayalam 90
- Maney, K. 209
- Manning, C.D. 46, 47
- Marathi 90
- mark-up 18–19, 71, 268–9
- Marten, L. 190
- Mason, I. 158, 159
- matching 37–40, 43, 80
- McKeown, K. 99
- Melby, A. 14, 29, 33, 47, 59, 64
- Metal* 9, 165, 190, 254
- Météo* 10, 136, 283–4, 289–95, 304
- metric 215, 216, 217, 230
- Meyer, I. 11, 63, 64
- MicroCat* 255
- Microsoft 163, 309
- Minnis, S. 242, 244
- minority languages 14, 87–102
non-indigenous, *see* NIML
- MIT 4
- Mitamura, T. 278, 279, 280, 281
- Mitkov, R. 335, 336
- Miyazawa, O. 329, 336
- Miyazawa, S. 208, 210
- MLV, *see* multi-language vendor
- morphology 91, 128, 145
- Moscow State University 331, 335
- Mowatt, D. 325, 336
- MT (Machine Translation) *passim*
- multi-language vendor (MLV) 83
- Munday, J. 107, 117
- N**
- Nagao, M. 158, 160, 228, 244
- Nakayama, K. 208, 210
- named entities 240
- Nasr, A. 277, 281
- Nazarenko, A. 63, 64
- Newton, J. 29, 30, 278, 281
- New York University 283
- Nida, E.S. 14 8, 160
- NIML (non-indigenous minority language) 88
- Nirenburg, S. 10, 11, 100, 102
- noisy channel 138
- Nomura, H. 242, 244
- normalisation 108, 113
- Norwegian 93, 109, 110
- Nyberg, E.H. 278, 279, 280, 281
- O**
- O'Brien, S. 52, 64, 322, 336
- O'Connell, T. 230, 236
- OCR 90, 94, 166
- office automation 220
- Ogden, C.K. 250, 281
- O'Hagan, M. 29, 30
- Olohan, M. 113, 117
- Olympic Games 293
- online
dictionaries 19
help 72
- OSCAR 34, 59, 63
- outbound, *see* dissemination
- Øverås, L. 109, 117
- OVUM 242, 300
- P**
- PACE (Perkins Approved Clear English)
250, 254, 255–6, 257
- Pan-American Health Organization
(PAHO) 299, 308
- Papineni, K. 240, 244
- partial match 32, 41
- Passolo 81, 82
- Pearson, J. 63, 65
- Pérez-Ortiz, J.A. 320, 336
- Perkins Engines Ltd 250, 254, 255, 278
see also PACE
- Polish 90

- Pontiero, G. 108
 Portuguese 108, 109, 112, 145–6, 164, 172, 180, 191, 204, 206
 post-editing 23, 220, 297–317, 322, 331
 automation of 313–14
 full 306
 minimal 304–6
 rapid (RPE) 302–3, 312, 313
 precision 252, 258
 pre-editing 23
 pre-translation 57
 probability 138, 139
 productivity 42, 186
 Punjabi 90
 Pym, P.J. 255–6, 278, 281
- Q**
 qualitative measure 215, 216
 quantitative measure 215, 216, 230
- R**
 radar chart 233–4
 RALI 29
 Raskin, V. 100, 102
 readability 248, 258
 Rebeyrolle, J. 63, 64
 recall 252, 258
 representation 124, 145, 146–50, 177, 332
 Reuther, U. 334, 337
 Richmond, I.M. 327–8, 337
 Ritchie, G.D. 158, 160
 Rogers, M. 63
 Rondeau, G. 58, 63, 65, 289
 round-trip translation 200, 202
 RPE, *see* post-editing
 Rudat, K. 88, 102
 rules 121, 128, 132, 135, 137, 176–7, 253, 292
 Rumanian 172
 Russian 172, 180, 228–9, 331
 -English 4, 164, 228–9
- S**
 SAE (Society for Automotive Engineering) 307, 316
 Sager, J.C. 63, 65, 294, 295
 Sager, N. 283
 Salkie, R. 111, 117
 Samuelsson-Brown, G. 29, 30
 Scandinavian languages 172
 ScaniaSwedish 251, 254
 Schmitz, K.-D. 63, 64, 65
 Schreurs, D. 278, 279
 Schubert, K. 334, 335, 336, 337
 Schütze, H. 46, 47
 Schwitter, R. 251, 280
 Scott, M. 108, 117
 Scott, N. 108, 117
 SE (Simplified English) 247, 249, 250, 251, 252, 254, 257–8, 278
 SECC 254, 277, 278
 Senez, D. 312, 307
 Serbian 90, 172
 SGML 19, 73, 259, 268
 Sheremeteva, S. 335, 337
 Shirai, S. 277, 281
 Shubert, S. 258, 281, 316, 317
 Siemens 251, 254
 simplification 107, 113, 178
 Simplified English, *see* SE
 simship 300
 Sinaiko, H.W. 242, 244
 Sinclair, J. 106, 117
 single-language vendor 84
 Slobin, D. 152, 160
 SLV, *see* single-language vendor
 Somali 90
 Somers, H.L. 10, 11, 29, 30, 46, 47, 89, 100, 102, 103, 141, 158, 159, 242, 243, 294, 295, 325, 334, 336, 337
Spanam 308
 Spanish 112, 129, 152, 164, 172, 180, 203–4, 206, 215–16, 225–6, 271–2
 -English 168, 172, 191, 202, 225–6, 236–8, 308–9
 sparse data 139
 specification 178
 speech recognition 16, 90, 213

- spell-checker 14–16, 90, 95, 98
spoken-language translation 7, 10, 162, 170
Stanley, J.C. 242, 243
statistical MT 138–40, 179
Steiner, E. 159, 160
stock-market reports 285
storage 53–4
Story, H. 208, 209, 194, 210
Strehlow, R.A. 53, 65
string-edit distance 38–9
subjectivity 214, 219, 228, 230, 238, 239,
 241
sublanguage 6, 181, 283–95
Swedish 93, 112, 251, 254
Sylheti 90
synthesis 123, 145
 problem 123, 133, 137
Systran [company] 9, 190, 191–206
Systran [system] 5, 164, 165, 169, 177, 191–
 206, 208, 302
- T**
T1 Professional 20–22, 190
Taber, C.R. 148, 160
tagger, tagging 60, 100
Talmy, L. 152, 153, 160
Tamil 90
TAUM 289
Taylor, K. 242, 238, 244
teaching 202, 319–39
 see also translator training
Telugu 90
tense 153–5
term 49
 bank 14, 20, 50, 51
 extraction 51, 60–2, 63
 record 50, 51–2, 53
termbase 51, 54, 57, 58
TermBase 54
terminology 14, 20, 34, 49, 97, 256, 263,
 271, 285
 lookup, automatic 56, 58, 81
 management system, *see* TMS
tools 49–65, 165, 169
Termium 14, 50
text type 182, 287
Thai 173
Thieroff, R. 158, 160
Thompson, A.D. 335, 337
Thompson, J. 335, 337
TM, *see* translation memory
TMS (Terminology Management System)
 50, 53–60, 81
TMX (Translation Memory eXchange) 34,
 45
tools 6, 14–16, 49–65, 304
Trados 22, 26, 31, 36, 46, 56, 57, 77, 80, 81
transfer 122, 123, 124, 134, 152, 153
 problem 123, 128–32, 137
TransIt 15, 80
TransIt TIGER 329
translation 69, 77, 90, 107, 119–42, 155
 divergences 151–3
 draft-quality 119
 ideal, perfect 213
 memory (TM) 14, 25, 31–46, 70, 72–3,
 80, 99, 166, 248
 studies, theory 106, 149
translational English 109, 113–4
Translation Manager 38
translator (human) 2, 115, 119–22, 148–9,
 152, 156, 168–9, 185–6, 188–9, 197, 213,
 216, 220, 270–2, 298, 303
 –’s workstation 6, 13–29, 161, 166–7
training 112–3, 319–39
trick sentences 321, 333
Trujillo, A. 10, 11, 141, 334, 337
Tsutsumi, T. 158, 160
- U**
Ulrych M. 112, 113, 117
Umino, B. 63, 64
UMIST 113, 334
unapproved 246, 252
Unicode 15, 68
uniformity 248
Université de Montréal 26, 29, 283, 288,
 289

-
- Université de Nancy 112
 University of Hull 329
 University of Leuven 254
 University of Limerick 80
 University of Oslo 109
 University of Zurich 251
 Urdu 90, 94, 98–9, 173
 Uren, E. 86
 user 184, 187–8, 196–203, 206, 220, 230–1,
 233, 239, 272, 329–31
 -friendliness 42
 interface 71
- V**
 van der Eijk, P. 99, 103, 278, 279, 281
 Vandooren, F. 151, 158, 160
 van Slype, G. 242, 244
 Vasconcellos, M. 63, 65, 242, 244, 308, 317
 Vauquois, B. 122, 141
 Veale, T. 297, 317
 Vertan, C. 335, 337
 v. Hahn, W. 335, 337
 Vietnamese 90, 172–3
 Vinay, J.-P. 152, 158, 160
- W**
 Wagner, E. 297, 311–12, 317
 Warburton, K. 63, 65
 Warwick-Armstrong, S. 99, 102
 Way, A. 297, 317, 335, 336, 337
 weather bulletins 286, 289–95
 Weaver, W. 4
- web-page, *see* World Wide Web
 Weidner, *see* MicroCat
 Welsh 90
 Westfall, E. 205, 210
 White, E.N. 250
 White, J.S. 230, 235, 236, 238, 242, 244
 Whitelock, P. 141, 142, 294, 295
 wildcard search 55
 Wojcik, R. 278, 279, 280, 281
 Wolfson College, Cambridge 250
 Woolls, D. 112, 116, 117
 word processing 14–15, 90, 92–4
WordSmith Tools 108, 112, 113
 workflow scenario 324–5
 WorldLanguage.com 89, 101
 World Wide Web 7, 20, 50, 68, 74, 82, 98,
 101, 113, 115, 170–1, 186, 191–210, 304,
 321, 325, 331
 writing systems 15
 Wright, S.E. 63, 64, 65
- X**
 XML 71, 73
 X-rated material 205
- Y**
 Yang, J. 178, 190, 206, 208, 210
- Z**
 Zanettin, F. 112, 117
 Zar, J. 96, 103
 Zhu, C. 158, 160