

TALN 2003

RECITAL 2003

Batz-sur-Mer
Du 11 au 14 juin 2003

Tome 2

Sous l'égide de
l'Association pour le Traitement Automatique des Langues (ATALA)

Sommaire général des Actes de TALN

Tome 1

Conférence d'ouverture
Actes de TALN
Posters de TALN
Actes de Récital
Posters de Récital

Tome 2

Tutoriels
Conférences associées

Sommaire

Tutoriels

Michael Carl – IAI <i>Introduction à la traduction guidée par l'exemple (Traduction par analogie)</i>	11
Didier Bourigault (1) et Nathalie Aussenac-Gilles (2) - (1) ERSS-CNRS Univ. Toulouse le Mirail, (2) IRIT Univ. Paul Sabatier <i>Construction d'ontologies à partir de textes</i>	27

Conférences associées

<i>Evaluation des analyseurs syntaxiques</i>	51
<i>Préface</i>	53
Salah Aït-Mokhtar, Caroline Hagège, Ágnes Sándor - XRCE <i>Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques</i>	57
Sophie Aubin - INRA <i>Evaluation comparative de deux analyseurs produisant des relations syntaxiques</i>	67
Philippe Blache (1) & Jean-Yves Morin (2) - (1) LPL-CNRS, Université de Provence, (2) Université de Montréal <i>Une grille d'évaluation pour les analyseurs syntaxiques</i>	77
V. Gendner, G. Illouz, M. Jardino, P. Paroubek, L. Monceaux, I. Robba, A. Vilnat - LIMSI <i>Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS</i>	87
Gil Francopoulo - Tagmatica <i>TagChunker : mécanisme de construction et évaluation</i>	95

TALN et multilinguisme	105
Luc Plamondon, George Foster - RALI - Université de Montréal <i>Multilinguisme et question-réponse: adaptation d'un système monolingue</i>	107
Ahmed Abdelali, James Cowie, David Farwell, Bill Ogden, and Stephen Helmreich <i>Cross-Language Information Retrieval using Ontology</i>	117
Olivier Kraif - LIDILEM <i>Repérage de traduction et commutation interlingue : Intérêt et méthodes</i>	127
Jacques Vergne - GREYC <i>Un outil d'extraction terminologique endogène et multilingue</i>	139
Bao-Quoc Ho, Jean-Pierre Chevallet, Marie-France Bruandet - CLIPS-IMAG <i>Mise en place d'un Système de Recherche d'informations en vietnamien</i>	149
Thi Minh Huyen Nguyen (1), Laurent Romary (1) and Xuan Luong Vu (2) - (1) LORIA, (2) Centre de Lexicographie du Vietnam <i>Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens</i>	161
Traitemen automatique des langues minoritaires et des petites langues	171
Atelach Alemu, Lars Asker and Mesfin Getachew - Addis Ababa University, Stockholm University <i>Natural Language Processing for Amharic: Overview and Suggestions for a Way Forward</i>	173
Sisay Fissaha , and Johann Haller - University of Saarland <i>Amharic verb lexicon in the context of Machine Translation</i>	183
Antônio Carlos da Rocha Costa and Graçaliz Pereira Dimuro - Universidade Católica de Pelotas <i>SignWriting and SWML: Paving the Way to Sign Language Processing</i>	193
Kevin P. Scannell - Saint Louis University <i>Automatic thesaurus generation for minority languages: an Irish example</i>	203
Victor Lascurain (1), Eneko Agirre (1), Mikel Lersundi (1), Luboš Popelínský (2) - (1) University of the Basque Country, Donostia, Spain, (2) Faculty of Informatics, Masaryk University <i>Disambiguation of case suffixes in Basque</i>	213

Caroline Gasperin(1), Renata Vieira(1), Rodrigo Goulart(1), Paulo Quaresma(2) - (1) PIPICA - Unisinos, (2) UEVORA <i>Extracting XML syntactic chunks from Portuguese corpora</i>	223
Oliver Streiter and Ernesto William De Luca - European Academy <i>Example-based NLP for Minority Languages: Tasks, Resources and Tools</i>	233
A. Diaz de Ilarrazo (1), A. Gurrutxaga (2), I. Hernaez (1), N. Lopez de Gereñu (3) and K. Sarasola (1) - (1) Ixa taldea - University of the Basque Country, (2) Elhuyar Fundazioa, (3) VicomTech <i>HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities</i>	243
Evelyn Bortolotti and Sabrina Rasom - ITC-IRST Trento Italia <i>Linguistic Resources and Infrastructures for the Automatic Treatment of Ladin Language</i>	253
Paul R. Bowden - The Nottingham Trent University <i>Building a Lexicon for a Kernewek MT System</i>	265
Oliver Streiter, Daniel Zielinski, Isabella Ties and Leonhard Voltmer - European Academy <i>Term Extraction for Ladin: An Example-based Approach</i>	275

<i>Index des auteurs</i>	287
---------------------------------	-----

TALN 2003

Tutoriels

Introduction à la traduction guidée par l'exemple (Traduction par analogie)

Michael Carl

Institut für Angewandte Informationsforschung,
Martin-Luther-Straße 14,
66111 Saarbrücken, Germany,
carl@iai.uni-sb.de

Mots-clefs – Keywords

traduction guidée par l'exemple, traduction par analogie, traduction statistique, induction de grammaire de traduction
example-based machine translation, analogical translation, statistical machine translation, induction of translation grammar

Résumé - Abstract

Le nombre d'approches en traduction automatique s'est multiplié dans les dernières années. Il existe entre autres la traduction par règles, la traduction statistique et la traduction guidée par l'exemple. Dans cet article je décris les approches principales en traduction automatique. Je distingue les approches qui se basent sur des règles obtenues par l'inspection des approches qui se basent sur des exemples de traduction. La traduction guidée par l'exemple se caractérise par la phrase comme unité de traduction idéale. Une nouvelle traduction est générée par analogie : seulement les parties qui changent par rapport à un ensemble de traductions connues sont adaptées, modifiées ou substituées.

Je présente quelques techniques qui ont été utilisées pour ce faire. Je discuterai un système spécifique, EDGAR, plus en détail. Je démontrerai comment des textes traduits alignés peuvent être préparés en termes de compilation pour extraire des unités de traduction sous-phrasiques. Je présente des résultats en traduction Anglais → Français produits avec le système EDGAR en les comparant avec ceux d'un système statistique.

In this paper I characterize a number of machine translation approaches: rule-based machine translation (RBMT), statistical machine translation (SMT) and example-based machine translation (EBMT). While RBMT systems make use of hand-build rules, SMT and EBMT systems explore and re-use a set of reference translations. EBMT systems are rooted in analogical reasoning, where the ideal translation unit is the sentence. Only if an identical sentence cannot be found in the reference material, EBMT systems modify, substitute and adapt sequences of the retrieved examples to generate a suitable translation.

I discuss runtime and compilation time techniques and I present a system, EDGAR, in more detail. I show how translation units are extracted off-line and how they are re-used during translation. The description of a series of experiments for the translation English → French conclude this paper. An extended bibliography provides further pointer for interested readers.

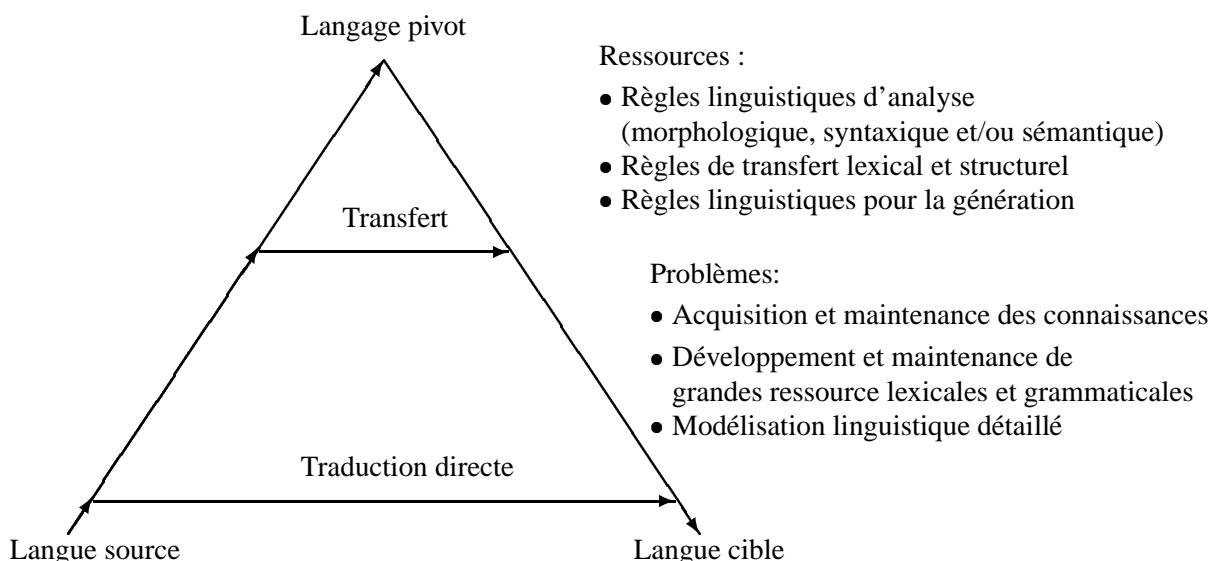
1 Contenu

- Caractérisation des approches à la traduction automatique
 - Traduction basée sur des règles, traduction statistique et traduction guidée par l'exemple
- Traduction guidée par l'exemple (EBMT)
 - Approches en termes de temps d'exécution et approches en termes de compilation
- Le système EDGAR
 - Segmentation, généralisation, spécification et raffinement
- Acquisition de grammaires de traduction
 - Propriétés de grammaires de traduction
 - Algorithme d'extraction de grammaire
- Expériences en traduction Anglais → Français
 - Acquisition des grammaires Anglais → Français à partir du Canadian Hansards
 - Echelonnement de l'approche
 - Comparaison avec *BabelFish* et traduction statistique
 - Intégration EBMT et SMT

2 Caractérisation des approches à la traduction automatique

2.1 Traduction automatique basée sur règles (RBMT)

Des approches en *traduction basée sur règles* (RBMT) sont fréquemment présentées par la pyramide de (Vauquois, 1968) (voir en dessous). Ces systèmes contiennent typiquement une série de fonctions qui analysent les phrases à traduire : analyses morphologiques, syntaxiques et/ou sémantiques, un module de transfert de la langue source en langue cible qui dépend du degré d'abstraction de la représentation du système, et une série de fonctions qui génèrent la phrase cible. Ces fonctions sont contrôlées par des dictionnaires et par des grammaires qui sont le plus souvent obtenues par l'inspection d'un (ou d'un groupe de) linguiste(s). Ceci a pour conséquence un développement lent du système principalement dû au problème d'acquisition de connaissances car les problèmes linguistiques de traduction doivent être d'abord complètement compris avant de les formuler en termes de règles. Mais beaucoup de problèmes en traduction automatique n'ont pas (encore) été entièrement compris ou requièrent une analyse complète sémantique et pragmatique ce qui n'est pas toujours disponible dans la plus part des cas.



2.2 Traduction basée sur des données

La *traduction basée sur des données* (Corpus-based Machine Translation (CBMT), mais aussi Data-driven Machine Translation) subsume un ensemble de méthodes alternatives et récentes qui visent à résoudre le problème de l'acquisition des connaissances en traduction par règles. Ces méthodes utilisent des textes bilingues traduits qui sont consultés lors de la traduction d'un texte ou d'une phrase nouvelle. Les textes bilingues sont alignés en segments de manière suivante:

Texte bilingue aligné (extrait du Canadian Hansard)

1	LA CHARTE CANADIENNE DES DROITS ET LIBERTÉS	canadian charter of rights and freedoms
2	L'hon. Benoît Bouchard (secrétaire d'État du Canada):	Hon. Benoît Bouchard (Secretary of State of Canada):
3	Monsieur le Président, je voudrais porter à l'attention de la Chambre que nous célébrons aujourd'hui, comme le savent les honorables députés, l'anniversaire de la proclamation de la Charte canadienne des droits et libertés qui a eu lieu le 17 avril 1982, ainsi que son parachèvement, il y a un an, avec l'entrée en vigueur des dispositions garantissant l'égalité à tous les membres de notre société.	Mr. Speaker, I would like to bring to the attention of the House that today, as Hon. Members are no doubt aware, we are celebrating the anniversary of the proclamation of the Canadian Charter of Rights and Freedoms which took place on April 17, 1982, and also of the coming into effect a year ago of the provisions guaranteeing equality for all members of our society.

Parmis le paradigme CBMT, deux directions principales peuvent être distinguées : la *traduction statistique* et la *traduction guidée par l'exemple*.

2.2.1 Traduction statistique (SMT)

La *traduction statistique* (SMT) se base sur la théorie mathématique de distribution et d'estimation probabiliste développée par Frederick Jelinek au IBM T.J. Watson Research Center et —en particulier— sur un article de (Brown et al., 1990). Les systèmes statistiques apprennent un modèle probabiliste de traduction ($Pr(t|s)$) à partir d'un texte bilingue et un modèle probabiliste de la langue cible ($Pr(t)$) à partir d'un texte monolingue. En temps d'exécution, la meilleure traduction pour une phrase nouvelle est recherchée grâce à la maximisation de ces deux modèles probabilistes.

$$\arg \max Pr(t|s) = \arg \max \{Pr(t) * Pr(s|t)\}$$

- modèle de traduction $Pr(s|t)$
 - modèle de langue $Pr(t)$
-
- Approche d'apprentissage non-supervisée basée sur les formes fléchies.
 - La traduction cible est synthétisée à partir de traduction(s) de mots individuels.
 - Grande quantité de textes bilingues alignés nécessaire pour l'entraînement.

Typiquement, RBMT et SMT génèrent la phrase cible à partir des traductions de mots simples et isolés. La ‘meilleure’ traduction est déterminée:

en SMT par les probabilités de $Pr(s|t)$ et $Pr(t)$
en RBMT par des contraintes exprimées par des règles

2.2.2 La traduction guidée par l'exemple (EBMT) : Traduction par Analogie

La *traduction guidée par l'exemple* (Example-Based Machine Translation, EBMT) prend sa place entre la RBMT et la SMT : beaucoup d'approches intègrent des règles et des techniques statistiques. Néanmoins il y a des caractéristiques qui distinguent l'EBMT de la SMT et de la RBMT :

- La ‘phrase’ est l’unité de traduction idéale.
- Traduction guidée par l’exemple consiste à :
 - rechercher les meilleurs exemple(s) de référence dans une base de données.
 - substituer, modifier et adapter des séquences différentes.

Beaucoup de techniques ont été utilisées et inventées en EBMT pour substituer, modifier et adapter les séquences de mots qui diffèrent dans les exemples de la base et les nouvelles phrases à traduire. Un excellent survol de ces techniques et de leur enjeu se trouve dans (Somers, 1999; Somers, 2003). Dans mon article je présente plus en détail :

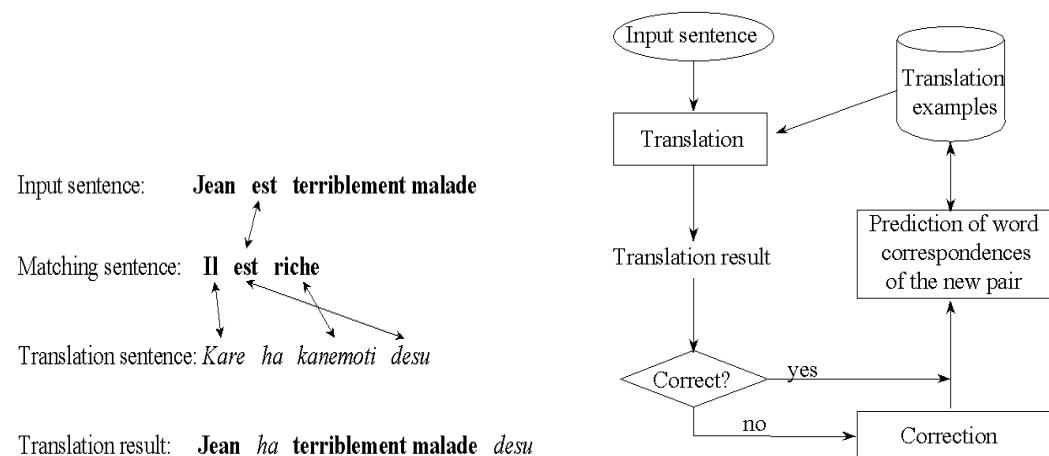
- Approches en termes de temps d’exécution
- Approches en termes de compilation
 - Représentations en schémas
 - Représentations en arbres syntaxiques

3 La traduction guidée par l'exemple (EBMT)

3.1 Approches en termes de temps d’exécution

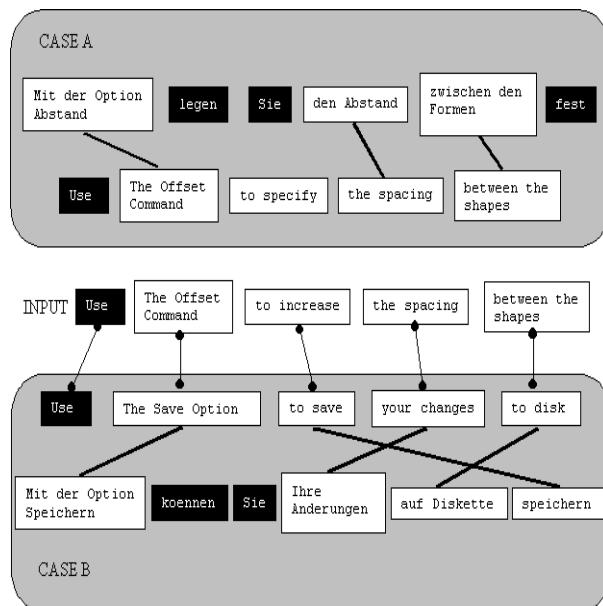
3.1.1 Segmentation dynamique

Dans l’approche proposé par (Andriamanankasina et al., 2003; Andriamanankasina et al., 1999) les exemples sont balisés et les correspondances entre les mots des deux phrases sont marqués. L’exemple le plus proche à la phrase nouvelle à traduire est recherché et des séquences égales sont traduites en langue cible. Ce processus est iteré jusqu’à ce que la phrase soit entièrement traduite où il n’y a plus d’exemple proche disponible dans la base. La traduction peut être corrigée manuellement et insérée dans la base de manière dynamique. Andriamanankasina *et al.* montrent que ce *cycles d’apprentissage* améliore les résultats de traduction obtenu.



3.1.2 Adaptabilité versus similarité en recherche

Dans le système de (Collins & Cunningham, 1997; Collins, 1998), voir aussi : (Collins & Somers, 2003), les exemples sont balisés et segmentés et portent l'information du rôle syntaxique. Les segments correspondants sont connectés d'une langue à l'autre. Le processus de recherche inclut une mesure d'adaptabilité qui indique la similarité de l'exemple par rapport à son contexte externe. La notion *adaptation-guided retrieval* (recherche guidée par l'adaptabilité) indique le degré auquel les exemples retrouvés sont un bon modèle pour la traduction désiré : alors que le "CASE A" est plus similaire du "INPUT", "CASE B" est le meilleur modèle pour sa traduction dû à sa meilleure adaptabilité.



3.2 Approches en termes de compilation

3.2.1 Extraction "linguistic-light"

Güvenir et Cicekli (Güvenir & Cicekli, 1998; Cicekli & Güvenir, 1996; Cicekli & Güvenir, 2003) présentent un algorithme pour l'extraction des correspondances lexicales de deux exemples de traduction : des parties du côté source doivent correspondre aux parties similaires du côté cible et des chaînes de mots différentes en langue source doivent correspondre à des chaînes de mots différentes en cible. Ces correspondances sont apprises en forme de *schémas de traduction* (translation template). Un schéma de traduction est un exemple de traduction généralisé dont certaines parties ont été remplacées par des variables liées.

Deux exemples de traduction :

<u>I took a</u>	ticket	from Mary	\leftrightarrow	Mary'den bir	bilet	<u>aldim</u>
<u>I took a</u>	pen	from Mary	\leftrightarrow	Mary'den bir	kalem	<u>aldim</u>

Généralisation de différences et extraction de correspondances lexicales :

$$\text{I took a } \mathcal{X}_1 \text{ from Mary} \leftrightarrow \text{Mary'den bir } \mathcal{Y}_1 \text{ aldim}$$

$$\begin{array}{ccc} \text{ticket} & \leftrightarrow & \text{bilet} \\ \text{pen} & \leftrightarrow & \text{kalem} \end{array}$$

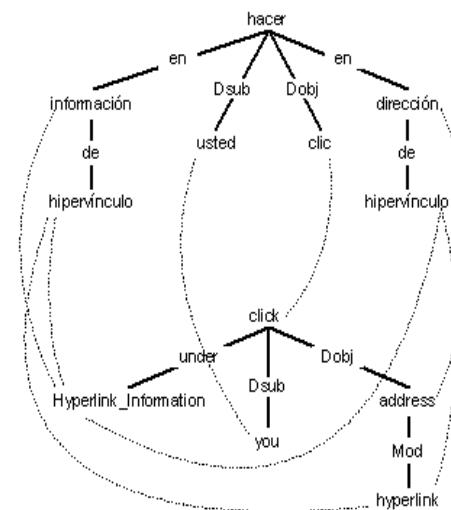
3.2.2 Extraction “linguistic-heavy” : Microsoft Research MT (MSR-MT)

(Richardson et al., 2001; Menezes & Richardson, 2003) utilisent des règles pour obtenir les formes logiques pour obtenir les formes logiques des exemples. Ces représentations sont connectées grâce à un lexique bilingue. Ensuite des connections ambiguës sont nettoyées avec des règles de préférence. Finalement des structures de transfert de haute qualité (ce qu'ils appellent des *transfer mappings*) sont extraites. Pour chaque structure de transfert la fréquence est calculée et un contexte suffisant est gardé pour distinguer les “mappings” ambiguës pendant la traduction.

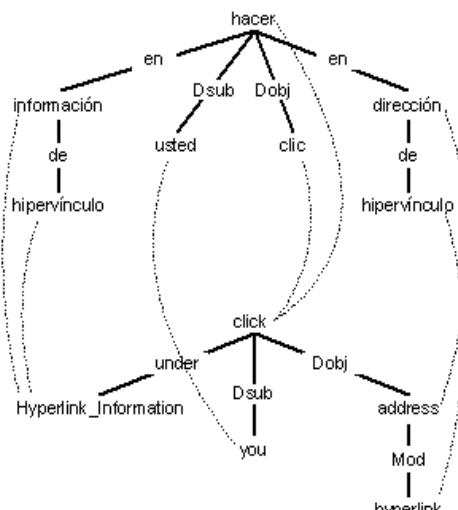
En Información del hipervínculo, haga clic en la dirección del hipervínculo.

Exemple de traduction: \longleftrightarrow
Under Hyperlink Information, click the hyperlink address.

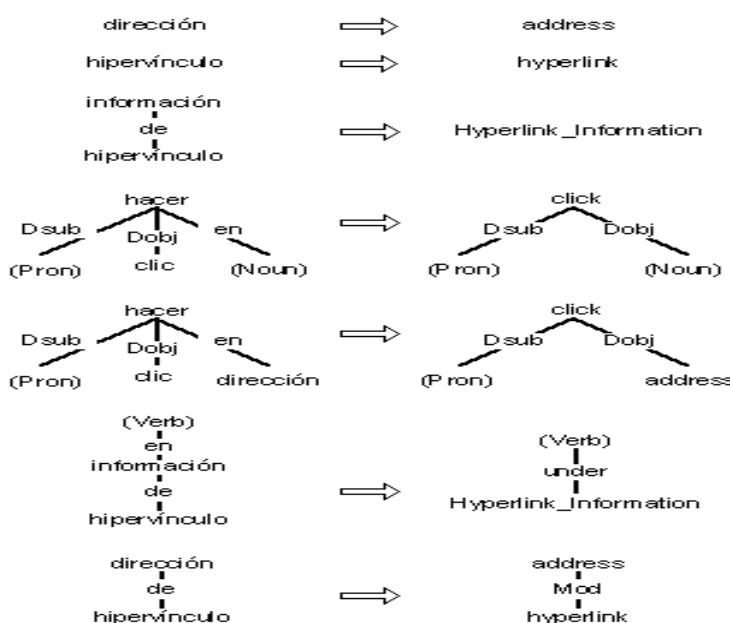
Correspondances lexicales entre les formes logiques (FL)



Alignement entre FL espagnol et anglais



Structures de transfert (transfer mappings) acquises de l'espagnol vers l'anglais



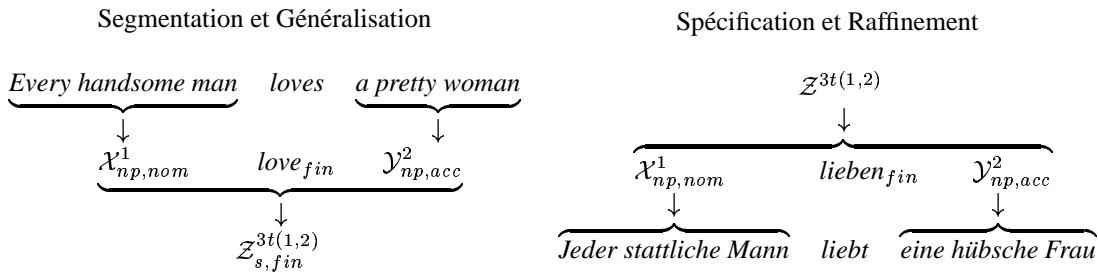
4 Présentation du système EDGAR

Le système EDGAR (Carl, 1999) utilise des analyseurs morphologique et syntaxique en plus des exemples de traduction. Un mécanisme d'induction généralise des exemples et produit une grammaire de traduction (Carl, 2003). La segmentation et généralisation d'une nouvelle phrase source ainsi que le raffinement de sa traduction dans la langue cible sont guidés par le contenu de la grammaire de traduction.

4.1 Segmentation, généralisation, spécification et raffinement

La grammaire de traduction contient des unités de traduction lexicales et des schémas de traduction variabilisés.

- 1 $(\text{Every handsome man})_{np} \longleftrightarrow (\text{Jeder stattliche Mann})_{np}$
- 2 $(\text{a pretty woman})_{np} \longleftrightarrow (\text{eine hübsche Frau})_{np}$
- 3 $(X_{np} \text{ love}_{fin} Y_{np})_s \longleftrightarrow (X_{np} \text{ lieben}_{fin} Y_{np})_s$



4.2 Représentation dans le programme EDGAR

Les entrées dans la grammaire portent l'information morphologique et les lemmas sous forme d'attribut/valeur, des traits. Les traits d'une analyse d'un mot peuvent être complexes (p.ex. `agr` en bas) ou atomiques (p.ex. `lu` en bas). De plus, les traits peuvent être atomique disjonctifs (p.ex. `case=d;g`) ou complexes disjonctifs. Par exemple, la représentation du mot allemand “der” porte les traits suivants:

$$\left\{ \begin{array}{l} \text{lu=d_art, c=w, sc=art, fu=def} \\ \text{agr=}\left\{ \begin{array}{l} \text{gen=f,} \\ \text{nb=sg,} \\ \text{case=d;g} \end{array} \right\}; \left\{ \begin{array}{l} \text{gen=m,} \\ \text{nb=sg,} \\ \text{case=n} \end{array} \right\}; \left\{ \begin{array}{l} \text{nb=plu,} \\ \text{case=g} \end{array} \right\} \end{array} \right\}, \left\{ \begin{array}{l} \text{lu=d_rel, c=w, sc=rel, fu=np,} \\ \text{agr=}\left\{ \begin{array}{l} \text{case=n,} \\ \text{g=m,} \\ \text{nb=sg} \end{array} \right\}; \left\{ \begin{array}{l} \text{case=g;g,} \\ \text{nb=sg,} \\ \text{g=f} \end{array} \right\} \end{array} \right\}$$

4.3 Percoler des traits avec des règles KURD

L'analyseur KURD (Carl & Schmidt-Wigger, 1998) sert à percoler les traits dans les arbres de dérivation et à unifier et substituer des valeurs dans les noeuds. La règle NP monte l'information d'accord des noeuds terminaux dans le noeud père.

```

NP = Aa {c=np} [
    e {c=w, sc=art, agr=_AGR},
    *a {c=adj, agr=_AGR},
    +e {c=noun, agr=_AGR}
]
: Au {agr=_AGR}

```

Les opérations de KURD

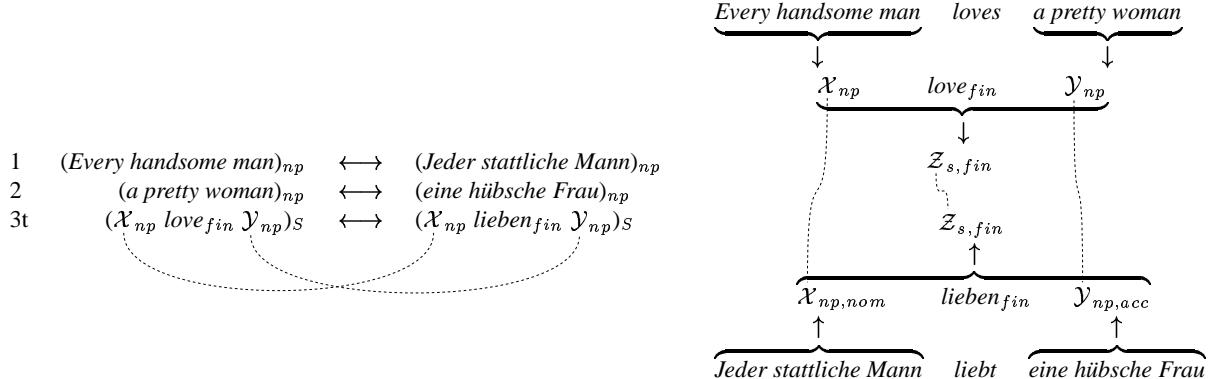
- Unification et suppression de traits.
- Concaténation et substitution de valeurs.
- Insertion et suppression de noeuds.

5 Acquisition de grammaires de traduction

5.1 Propriétés des grammaires de traduction

5.1.1 Grammaire de traduction homomorphe et arbres de dérivation isomorphes

Les grammaires *homomorphes* produisent des arbres *isomorphes* et rendent possible un transfert 1-à-1 de la source à la cible.



5.1.2 Traduction compositionnelle versus non-monotone (partiellement compositionnelle)

Les grammaires *compositionnelles* segmentent la phrase source récursivement en expressions qui sont traduites indépendamment tandis que les grammaires non-monotones s'arrêtent à un certain point.

	Exemple	Grammaire
compositionnelle:	business trip \leftrightarrow viaje de negocios	$X_{noun} \ Y_{noun}$ \leftrightarrow Y_{noun} de X_{noun} business \leftrightarrow negocios trip \leftrightarrow viaje
non-compositionnelle:	field trip \leftrightarrow viaje de estudio	field trip \leftrightarrow viaje de estudio
non-monotone	long field trip \leftrightarrow viaje de estudio largo	long \leftrightarrow largo field trip \leftrightarrow viaje de estudio $X_{adj} \ Y_{noun}$ \leftrightarrow $Y_{noun} \ X_{adj}$

5.1.3 Grammaire ambiguë versus inverse

Les grammaires *ambiguës* permettent plusieurs traductions pour une expression source tandis que les grammaires *inverses* ne produisent qu'une seule traduction.

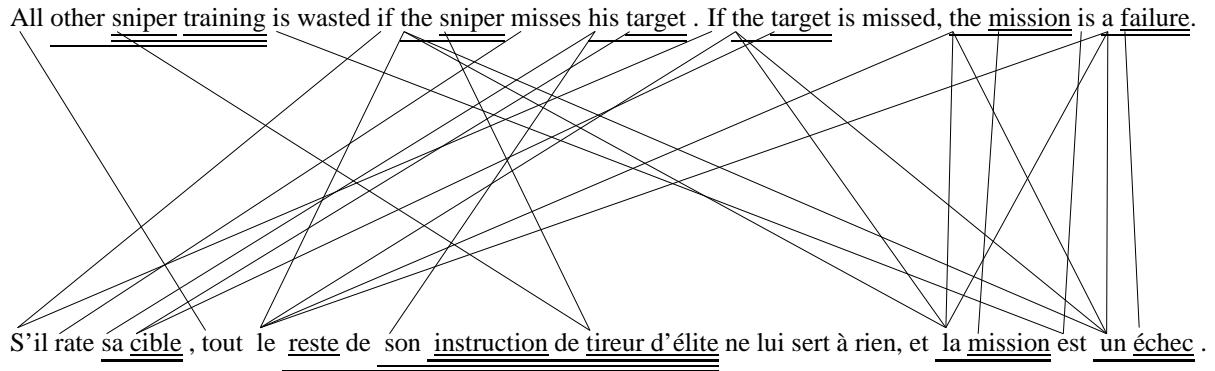
Grammaire ambiguë :	Grammaire inverse :
$X_{noun} \ Y_{noun}$ \leftrightarrow Y_{noun} de X_{noun} business \leftrightarrow negocios trip \leftrightarrow viaje field \leftrightarrow estudio field \leftrightarrow campo studies \leftrightarrow estudio	$X_{noun} \ Y_{noun}$ \leftrightarrow Y_{noun} de X_{noun} business trip \leftrightarrow viaje de negocios business \leftrightarrow negocios trip \leftrightarrow viaje field trip \leftrightarrow viaje de estudio

5.2 Extraction de grammaire de traduction : un algorithme

L'extraction de grammaires à partir d'exemples de traduction se poursuit en quatre étapes.

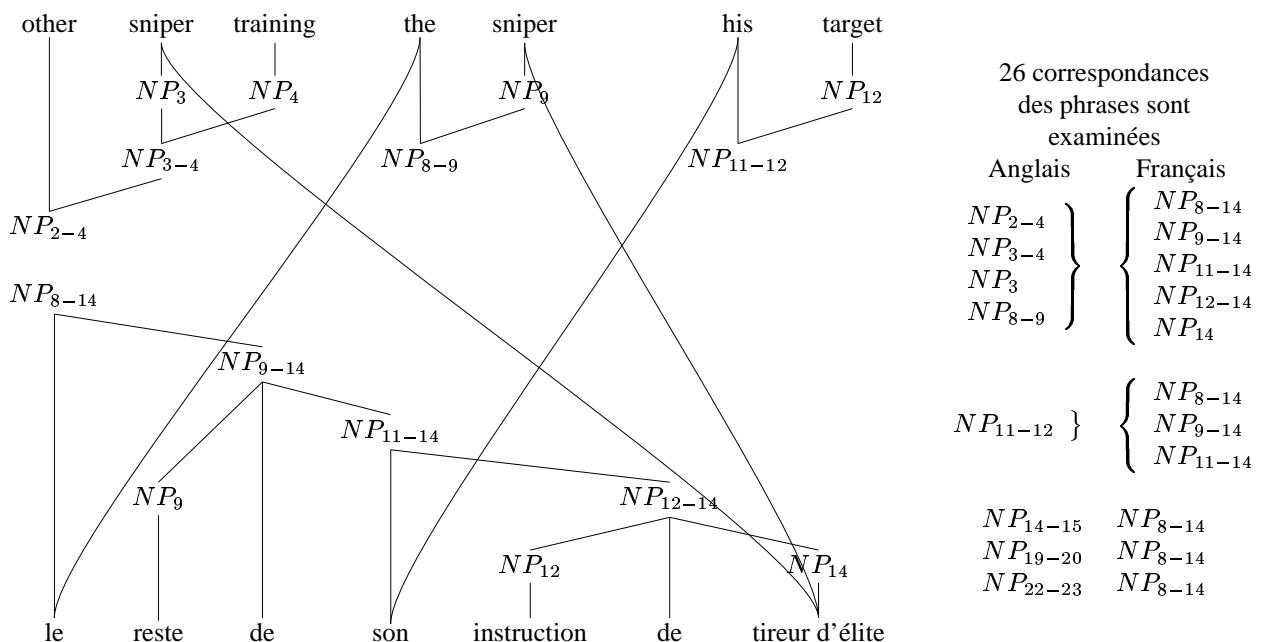
5.2.1 Alignement partiellement analysé et ancré avec un dictionnaire bilingue

D'abord l'analyse syntaxique partielle des deux côtés de l'alignement est effectuée. Des lexèmes des deux côtés sont connectés grâce à un dictionnaire bilingue.



5.2.2 Détermination de correspondances phrase à phrase les plus significantes

Les poids des correspondances des arbres sont calculés à partir (i) des poids et le nombre des ancrages lexicaux (ii) la fréquence des correspondances dans le texte et (iii) l'isomorphisme des analyses partielles (cf. (Carl, 2003))



Cet extrait d'alignement de la section 5.2.1 montre trois segments anglais différents connectés avec un segment français. Sont examinés 26 correspondances de phrases possibles dont la traduction $NP_{2-4} \leftrightarrow NP_{9-14}$: "other sniper training \leftrightarrow reste de son instruction de tireur d'élite" est détectée la plus consistante dans le texte aligné.

5.2.3 Schémas de traduction générés

Les schémas sont générés par la substitution des correspondances compositionnelles.

All NP_{2-4} is wasted if the sniper misses NP_{11-12} . If the target is missed, NP_{20-21} is NP_{23-24} .

S'il rate NP_{4-5} , tout le NP_{9-14} ne lui sert à rien, et NP_{22-23} est NP_{25-26} .

5.2.4 Grammaire de traduction générée

Une grammaire de traduction *compositionnelle* et *homomorphe* est extraite récursivement pour chaque exemple de traduction:

1 All other sniper training is wasted if the sniper misses his target.

If the target is missed, the mission is a failure.

\leftrightarrow

S'il rate sa cible tout le reste de son instruction de tireur d'élite ne lui sert à rien,

et la mission est un échec.

2 All NP^1 is wasted if the sniper misses NP^2 . If the target is missed, NP^3 is NP^4 .

\leftrightarrow S'il rate NP^2 tout le NP_1 ne lui sert à rien, et NP^3 est NP^4 .

3 other sniper training \leftrightarrow reste de son instruction de tireur d'élite

4 other NP^1 NP^2 \leftrightarrow reste de son NP^2 de NP^1

5 training \leftrightarrow instruction

6 sniper \leftrightarrow tireur d'élite

7 his target \leftrightarrow sa cible

8 his NP^1 \leftrightarrow sa NP^1

9 the mission \leftrightarrow la mission

10 the NP^1 \leftrightarrow la NP^1

11 mission \leftrightarrow mission

12 a failure \leftrightarrow un échec

13 a NP^1 \leftrightarrow un NP^1

6 Expériences en traduction guidée par l'exemple

Dans cette section nous générerons des grammaires avec l'approche présenté en section 5. Un texte test est traduit avec EDGAR présenté en section 4. Une description plus étendue des expériences peut être trouvé dans (Carl & Langlais, 2003). Les expériences se basent sur le *Canadian Hansards*, texte bilingue Anglais \leftrightarrow Français. Nous présentons des expériences différentes d'extraction de grammaire aussi bien en ce qui concerne le nombre d'exemples que le degré d'ambiguïté de la grammaire générée.

6.1 Extraction d'une grammaire de traduction

Les ressources utilisées pour extraire une grammaire de traduction (GT_1) Anglais \leftrightarrow Français incluent un dictionnaire bilingue de 77.016 entrées, un programme de segmentation en segments et un ensemble de 50.000 exemples de traduction alignés du Canadian Hansard. Le dictionnaire couvre prèsque 3/4 des mots anglais et français du ET₁ mais contient seulement à peu près 1/3 des mots différents qui occurent dans les deux textes. Le programme de segmentation (parser partiel) génère au moyen 11 et 13 segments

La traduction guidée par l'exemple

rsp. par exemple de traduction pour l'anglais et le français. Plus que la moitié des segments différents (63% et 50% rsp.) font partie des règles lexicales de la grammaire inverse extraite.

	Anglais	Français		Anglais	Français
Exemples de Traduction ET₁ :					
#exemples	50.000	50.000	Dictionnaire bilingue (DIC) :		
#mots	888.018	947.194	#entrées du dictionnaire	77.016	77.016
#mots different	17.915	23.675	%couvert en ET ₁	74,77%	74,99%
#mots/exemples	17,76	18,94	#mots different	7.688	7.714
			%anchors (en ET ₁)	42,28%	43,53%
Grammaire de traduction extraite GT₁ :					
#règles lexicales		113.810	Segments générés par le parser partiel :		
#schémas de traduction		70.153	#segments	581,599	650,136
			#segments différents	180,006	226,339

6.2 Traduction d'un texte test (TT)

Un texte test de 500 phrases est traduit de l'anglais vers le français avec les règles lexicales de GT₁. Alors que la couverture du dictionnaire bilingue (DIC) est plus grande que celle de la grammaire GT₁, la qualité de traduction, mesurée en WER¹ et en BLEU (Papineni et al., 2002), est mieux en traduction GT₁. Nous voyons ici une corrélation entre la qualité des traductions et la longueur des segments utilisés lors de la traduction.

	Texte Test (TT)		GT ₁	DIC	
	Anglais	Français	% mots couverts	66,38%	66,99%
#exemples	500	500	BLEU	0,1421	0,0573
#mots	8.665	9.806	WER	68,89%	81,68%
#mots/exemples	17,33	19,61	longueur segments ≥ 2		
			#segments	966	146
			#mots couverts	2,652	325
			%mots couverts	30,61%	3,75%

6.3 Echelonnement de grammaires inverses et ambiguës

Dans cette expérience nous étudions (i) la capacité de l'algorithme d'utiliser un nombre différent d'exemples de traduction et (ii) l'effet de l'utilisation d'unités ambiguës. Nous comparons trois grammaires différentes générées à partir d'ensembles différents d'exemples de traduction , tous extraits du Canadian Hansards.

	ET ₀	ET ₁	ET ₂
#exemples de traduction	10.000	50.000	100.000
#mots anglais (En)	151.954	888.018	1.437.450
#mots français (Fr)	163.113	947.194	1.503.196
#mots différents En	7.343	17.915	22.501
#mots différents Fr	9.528	23.675	29.559

Ces trois ensembles de référence sont utilisés afin de générer deux types de grammaires différentes : des grammaires inverses GT₀, GT₁ et GT₂ (dont GT₁ est égale à celle des sections 6.1 et 6.2) et des grammaires ambiguës GT₀^a, GT₁^a et GT₂^a. Les grammaires ambiguës contiennent près de 20% plus de règles de transfert lexical, alors que le nombre de mots différents reste à peu près pareil dans les deux

¹Les chiffres WER supérieures et chiffres BLEU inférieurs indiquent le meilleur résultat de traduction.

types de grammaires. On observe aussi que le nombre moyen de mots par règle augmente dans les grammaires plus grandes.

	Règles inverses de transfert lexical			Règles ambiguës de transfert lexical		
	GT ₀	GT ₁	GT ₂	GT ₀ ^a	GT ₁ ^a	GT ₂ ^a
#règles lexicales	23.214	113.810	180.745	28.393	146.684	220.248
#mots Anglais (En)	203.426	1.223.260	1.856.392	222.473	1.355.331	2.030.390
#mots Français (Fr)	220.273	1.314.197	2.007.322	244.615	1.491.559	2.219.455
#mots différents En	7.338	17.910	22.488	7.340	17.914	22.495
#mots différents Fr	9.520	23.659	29.523	9.521	23.670	29.542

En ce qui concerne la qualité des traductions produites, les deux types de grammaire produisent un taux de WER et BLEU à peu près égal. Ceci alors qu'un nombre considérable de segments de longueur supérieure à été utilisé pour produire la traduction avec des grammaires ambiguës. Nous concluons que les entrées ambiguës représentent pour la plupart des unités de traduction de qualité inférieure.

	Qualité du texte test en grammaires inverses ...			et qualité en grammaires ambiguë		
	GT ₀	GT ₁	GT ₂	GT ₀ ^a	GT ₁ ^a	GT ₂ ^a
WER	71,91%	68,89%	66,93%	71,88%	69,75%	67,22%
BLEU	0,1365	0,1421	0,1704	0,1398	0,1519	0,1706
#segments	3.581	4.405	4.685	3.599	4.314	4.659
#segments différents	992	1.279	1.387	1.055	1.343	1.450
#mots couverts	4.611	5.752	6.146	4.680	5.752	6.228
#segments longueur ≥ 2	767	966	1.050	816	1.032	1.108
#segments different	353	519	589	380	570	646
#mots couverts	1.952	2.652	2.863	2.170	2.844	3.062

6.4 Comparaison de GT, SMT et *BabelFish*

Dans cette expérience nous comparons les résultats de traduction obtenus utilisant les grammaires GT_{0–2} avec un système statistique (Langlais, 2002) entraîné sur les mêmes exemples de traduction ET_{0–2}. Nous voyons que les résultats SMT sont inférieurs (toujours WER et BLEU) à ceux obtenus en GT. Le système SMT₃ qui a été entraîné sur un texte de 1,5 millions de exemples (15 fois plus que ET₂) obtient les meilleurs résultats.

score	GT ₀	GT ₁	GT ₂	SMT ₀	SMT ₁	SMT ₂	SMT ₃	<i>BabelFish</i>
BLEU	0,1365	0,1421	0,1704	0,1156	0,1231	0,1378	0,2061	0,1578
WER	71,91%	68,89%	66,93%	74,72%	73,54%	71,52%	61,66%	66,03%

Le système commercial *BabelFish* obtient des résultats inférieurs à ceux de SMT₃ et GT₂. Ceci est surtout dû aux particularités du texte traduit : alors que GT et SMT apprennent les traductions particulières, *BabelFish* n'a pas pu être adapté à ce type de texte. Ainsi, la traduction : “the speaker \leftrightarrow le président” a été réalisée par GT et SMT alors que *BabelFish* génère la traduction “le haut-parleur”. De même : “some hon. members : oh , oh ! \leftrightarrow des voix : oh , oh !” est une traduction qui se voit fréquemment en Canadian Hansards mais *BabelFish* produit “membres d'un certain hon : l' OH OH”. Alors que ce sont des traductions possibles correctes dans d'autres contextes, elles sont erronées quant à la traduction du Canadian Hansards.

6.5 Intégration SMT et GT

Finalement nous essayons d'intégrer les grammaires GT et le système statistique suivant (Langlais, 2002) : quand la grammaire GT contient une entrée égale à une séquence de mots dans la phrase à traduire, le système SMT est forcé d'intégrer la traduction proposée par GT dans sa sortie. La qualité produite du système hybride est meilleure quant aux grammaires inverses ($SMT_{0-2}-GT_{0-2}$); pour l'intégration des grammaires ambiguës dans le système statistique ($SMT_{0-2}-GT_{0-2}^a$) une amélioration des résultats n'a pas pu être observé.

	SMT_0-GT_0	SMT_1-GT_1	SMT_2-GT_2	$SMT_0-GT_0^a$	$SMT_1-GT_1^a$	$SMT_2-GT_2^a$
BLEU	0.1495	0.1684	0.1789	0.1406	0.1541	0.1654
WER	71.19%	70.32%	68.94%	72.70%	72.45%	71.41%

7 Résumé et conclusion

Dans cet article je présente d'approches en traduction automatique. Je fais la distinction entre la traduction par règles (RBMT), la traduction statistique (SMT) et la traduction guidée par l'exemple (EBMT). Les ressources nécessaires en RBMT sont obtenues par l'inspection d'un (ou d'un groupe de) linguiste(s), tandis que les approches EBMT et SMT extraient les connaissances de traduction à partir des textes bilingues alignés. Au contraire à la SMT, la 'phrase' est l'unité de traduction idéale en EBMT. Je présente des systèmes EBMT qui extraient et acquièrent ces unités en termes de temps d'exécution et en termes de temps de compilation.

Ensuite je discute plus en détail le système EDGAR. A partir des exemples de traduction, EDGAR produit la traduction des phrases nouvelles par analogie de manière compositionnelle et isomorphe. Je présente un algorithme pour extraire une grammaire de traduction à partir des exemples de traduction. L'article conclut avec la description d'une série de expériences en traduction guidée par l'exemple. De ces expériences nous concluons que :

- La couverture de la grammaire est fonction du nombre des exemples de référence.
- Les grammaires produisent une meilleure qualité de traduction que la traduction SMT (taille identique de référence)
- Les règles ambiguës n'améliorent pas la qualité de la traduction.
- L'intégration des techniques EBMT et SMT améliore les résultats de la traduction.

References

- Al-Adhaileh, M. H. & Tang E. K. 1999. Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema. *Machine Translation Summit VII*, Singapore, 244–249.
- Andriamanankasina, T., K. Araki, & K. Tochinai. 2003. Ebmt of pos-tagged sentences with inductive learning. In (Carl & Way, 2003).
- Andriamanankasina, T., K. Araki & K. Tochinai. 1999. Example-Based Machine Translation of Part-Of-Speech Tagged Sentences by Recursive Division. *Machine Translation Summit VII*, Singapore, 509–517.
- Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer & P. S. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics* **16**, 79–85.
- Brown, R. D. 1996. Example-Based Machine Translation in the Pangloss System. *Coling* (1996), 169–174.

- Brown, R. D. 1997. Automated Dictionary Extraction for “Knowledge-Free” Example-Based Translation. *TMI* (1997), 111–118.
- Brown, R. D. 1999. Adding Linguistic Knowledge to a Lexical Example-based Translation System. *TMI* (1999), 22–32.
- Carl, M. & A. Way, editors. 2003. *Recent Advances in Example-Based Machine Translation*. Kluwer, Academic Publisher, Boston/Dordrecht/London. in press.
- Carl, M. 2003. Inducing translation grammars from bracketed alignments. In (*Carl & Way, 2003*).
- Carl, M. & Langlais, P. 2003. Tuning general purpose translation knowledge to a sublanguage. In *Proceedings of EAMT/CLAW*.
- Carl, M. 1999. Inducing Translation Templates for Example-Based Machine Translation. *Machine Translation Summit VII*, Singapore, 250–258.
- Carl, M. & Schmidt-Wigger, A.. 1998. Shallow Postmorphological Processing with KURD. In *Proceedings of NeMLaP3/CoNLL98*, pages 257–265, Sydney.
- Cicekli, I. & H.A. Güvenir. 2003. Learning Translation Templates from Bilingual Translation Examples. In (*Carl & Way, 2003*).
- Cicekli, I. & H. A. Güvenir. 1996. Learning Translation Rules From A Bilingual Corpus. *NeMLaP-2: Proceedings of the Second International Conference on New Methods in Language Processing*, Ankara, Turkey, 90–97.
- Collins, B. & H. Somers. 2003. EBMT Seen as Case-based Reasoning. In (*Carl & Way, 2003*).
- Collins, B. 1998. *Example-Based Machine Translation: An Adaptation-Guided Retrieval Approach*. PhD thesis, Trinity College, Dublin.
- Collins, B. & P. Cunningham. 1997. Adaptation Guided Retrieval: Approaching EBMT with Caution. *TMI* (1997), 119–126.
- Cranias, L., H. Papageorgiou & S. Piperidis. 1994. A Matching Technique in Example-Based Machine Translation. *Coling* (1994), 100–104.
- Furuse, O. & H. Iida. 1992. An Example-Based Method for Transfer-Driven Machine Translation. *TMI* (1992), 139–150.
- Furuse, O. & H. Iida. 1994. Constituent Boundary Parsing for Example-Based Machine Translation. *Coling* (1994), 105–111.
- Güvenir, H. A. & I. Cicekli. 1998. Learning Translation Templates from Examples. *Information Systems* **23**, 353–363.
- Kaji, H., Y. Kida & Y. Morimoto. 1992. Learning Translation Templates from Bilingual Text. *Coling* (1992), 672–678.
- Langlais, P. 2002. Ressources terminologiques et traduction probabiliste: premiers pas positifs vers un système adaptatif. In *TALN-2002*.
- Matsumoto, Y. & M. Kitamura. 1995. Acquisition of Translation Rules from Parallel Corpora. In R. Mitkov & N. Nicolov (eds) *Recent Advances in Natural Language Processing: Selected Papers from RANLP’95*, Amsterdam: John Benjamins, 405–416.
- McTait, K. & A. Trujillo. 1999. A Language-Neutral Sparse-Data Algorithm for Extracting Translation Patterns. *TMI* (1999), 98–108.

La traduction guidée par l'exemple

- Menezes, A. & S.D. Richardson. 2003. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In (*Carl & Way, 2003*).
- Meyers, A., R. Yangarber, R. Grishman, C. Macleod & A. Moreno-Sandeval. 1998. Deriving Transfer Rules from Dominance-Preserving Alignments. *Coling-ACL* (1998), 843–847.
- Nagao, M. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn and R. Banerji (eds) *Artificial and Human Intelligence*, 173–180, Amsterdam: North-Holland.
- Nirenburg, S., S. Beale & C. Domashnev. 1994. A Full-Text Experiment in Example-Based Machine Translation. *International Conference on New Methods in Language Processing (NeMLaP)*, Manchester, England, 78–87.
- Papineni, K., S. Roukos, T. Ward, & W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, Pennsylvania, USA, 311–318.
- Richardson, S.D., W.B. Dolan, A. Menezes & J. Pinkham. 2001. Achieving Commercial-quality Translation with Example-based Methods. *MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, 293–298.
- Sato, S. & M. Nagao. 1990. Toward Memory-Based Translation. *Coling* (1990), Vol. 3, 247–252.
- Somers, H. 1999. Review Article: Example-based Machine Translation. *Machine Translation*, 14(2):113–157.
- Somers, H. 2003. An Overview of EBMT. In (*Carl & Way, 2003*).
- Sumita, E. & H. Iida. 1991. Experiments and Prospects of Example-Based Machine Translation. *29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, 185–192.
- Sumita, E., H. Iida & H. Kohyama. 1990. Translating with Examples: A New Approach to Machine Translation. *TMI* (1990), 203–212.
- Vauquois, B. 1968. A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation. *IFIP Congress-68*, Edinburgh, 254–260; reprinted in Ch. Boitet (ed.) *Bernard Vauquois et la TAO: Vingt-cinq Ans de Traduction Automatique – Analectes*, 201–213, Grenoble (1988): Association Champollion.
- Veale, T. & A. Way. 1997. *Gaijin*: A Bootstrapping Approach to Example-Based Machine Translation. *International Conference, Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, 239–244.
- Watanabe, H. 1992. A Similarity-Driven Transfer System. *Coling* (1992), 770–776.
- Watanabe, H. 1993. A Method for Extracting Translation Patterns from Translation Examples. *TMI* (1993), 292–301.
- Watanabe, H. & K. Takeda. 1998. A Pattern-Based Machine Translation System Extended by Example-Based Processing. *Coling-ACL* (1998), 1369–1373.
- Way, A. 1999. A Hybrid Architecture for Robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence* 11, 441–471.

Construction d'ontologies à partir de textes

Didier Bourigault (1) et Nathalie Aussénac-Gilles

(1) ERSS – CNRS & Université Toulouse le Mirail

5, allées Antonio Machado

31 058 Toulouse Cedex 1

didier.bourigault@univ-tlse2.fr

(2) IRIT – Université Paul Sabatier

118, route de Narbonne, 31062 Toulouse Cedex 4

[aussénac@irit.fr](mailto:aussenac@irit.fr)

Résumé – Abstract

Cet article constitue le support d'un cours présenté lors de la conférence TALN 2003. Il défend la place du Traitement Automatique des Langues comme discipline clé pour le développement de ressources termino-ontologiques à partir de textes. Les contraintes et enjeux de ce processus sont identifiés, en soulignant l'importance de considérer cette tâche comme un processus supervisé par un analyste. Sont présentés un certain nombre d'outils logiciels et méthodologiques venant de plusieurs disciplines comme le TAL et l'ingénierie des connaissances qui peuvent aider l'analyste dans sa tâche. Divers retours d'expérience sont présentés.

This paper gathers the notes of a tutorial. We advocate in favour of the role of Natural Language Processing as a key discipline for the development of terminological and ontological resources from texts. The constraints and challenges of this process are identified, and lead to underline this task as a supervised processes carried out by an analyst. We present several software and methodological tools from NLP and knowledge engineering that can be used for to assist the analyst. Our suggestion rely on various experience feed-back.

Keywords – Mots Clés

Extraction de termes, extraction de relations, Terminologie, Ontologies, Ingénierie des connaissances, méthode, modélisation de connaissances, interdisciplinarité.

Term extraction, relation extraction, Terminology, ontologies, Knowledge Engineering, method, knowledge modelling, crossdisciplinarity.

1 Introduction

Dans cet article, nous développons les grandes lignes du cours présenté lors de la dixième conférence *Traitement Automatique des Langues* le 14 juin 2003 à Batz-sur-Mer. Ce cours fait suite aux tutoriels donnés en juin 2000 lors de la conférence *Ingénierie des connaissances* (IC 2000, Toulouse) et en janvier 2002 lors de la conférence *Reconnaissance des Formes et Intelligence Artificielle* (RFIA 2002, Angers), au cours desquels nous avons eu l'occasion de présenter les outils développés en Traitement Automatique des Langues (TAL) aux membres de la communauté d'*Ingénierie des Connaissances* (IC). L'objectif du présent cours est, symétriquement, de présenter aux chercheurs de la communauté Traitement Automatique des Langues les enjeux, pratiques et théoriques, l'utilisation de certains outils de TAL dans une perspective d'*ingénierie des connaissances*, ceci pour encourager les travaux interdisciplinaires autour de la problématique de construction de ressources termino-ontologiques (RTO)¹, à partir de textes. Cette problématique constitue en effet un nouvel enjeu important aussi bien pour le Traitement Automatique des Langues que pour l'*Ingénierie des Connaissances*. Les systèmes de traitement de l'information qui doivent fonctionner dans des domaines de connaissances spécialisés ne peuvent être efficaces que s'ils s'appuient sur des ressources termino-ontologiques, construites pour le domaine et l'application concernés. Les recherches et réalisations en TAL et en IC doivent être menées de façon pluridisciplinaire, pour, d'une part, développer les outils de TAL pertinents pour la tâche de construction de RTO à partir de textes, et, d'autre part, élaborer des méthodes d'acquisition des connaissances à partir de textes qui spécifient comment utiliser les outils de TAL et les environnements de modélisation des connaissances dans le contexte de la construction de RTO. Au delà, mais cet aspect sera moins développé dans ce cours, il s'agit de s'interroger sur le statut de la langue écrite comme révélateur de connaissances, dès lors que l'on veut y accéder au moyens d'outils informatiques.

2 Des ressources à construire variées, des outils génériques

2.1 RTO et systèmes de traitements de l'information

A la suite à l'utilisation généralisée des outils de bureautique, à l'internationalisation des échanges et au développement d'Internet, la production de documents sous forme électronique s'accélère sans cesse. Or pour produire, diffuser, rechercher, exploiter et traduire ces documents, les systèmes de gestion de l'information ont besoin de ressources termino-ontologiques, qui décrivent les termes et les concepts du domaine, selon un mode propre au type de traitement effectué par le système. La gamme des ressources à base terminologique et ontologique est aussi large que celle des systèmes de traitement de l'information utilisés dans les entreprises et dans les institutions :

- bases de données terminologiques multilingues classiques pour l'aide à la traduction,

¹ Dans ce cours, nous nous efforcerons d'utiliser systématiquement cette expression plutôt que le terme très en vogue d'*ontologie*, adopté dans le titre du cours pour des raisons de concision. Ce choix terminologique sera justifié plus loin dans cet article.

- thesaurus pour les systèmes d'indexation automatique ou assistée, index hypertextuels pour les documentations techniques,
- terminologies de référence pour les systèmes d'aide à la rédaction,
- référentiels terminologiques pour les systèmes de gestion de données techniques,
- ontologies pour les mémoires d'entreprise, les systèmes d'aide à la décision ou les systèmes d'extraction d'information,
- ontologies pour le Web sémantique,
- glossaires de référence, liste de termes pour les outils de communication interne et externe,
- etc.

Du côté de la recherche, chacune de ces ressources est prise en charge par une discipline différente. La terminologie focalise ses recherches, depuis l'avènement des outils de bureautique, sur les bases de données terminologiques destinée aux traducteurs humains. Les sciences de l'information et de la documentation concentrent leurs réflexions sur les thesaurus et langages de classification ou langages documentaires, exploités par les documentalistes pour indexer et classer les éléments de fonds documentaire. En informatique, le domaine de la recherche d'information (RI) s'intéresse à des thesaurus d'un type différent, conçus pour limiter le bruit et augmenter le rappel des outils informatiques de recherche d'information. L'intelligence Artificielle et l'Ingénierie des Connaissances travaillent sur les ontologies formelles qui constituent le cœur des systèmes à base de connaissances. Ces différentes disciplines développent de façon relativement autonome et cloisonnée des recherches spécifiques sur ces différents types de ressources. Or, sous la pression des besoins et des applications, elles sont amenées à considérer que, pour des raisons de pertinence et d'efficacité, les ressources lexicales et/ou conceptuelles qu'elles doivent construire et exploiter peuvent ou doivent être construites à partir de sources textuelles. Elles sont donc naturellement amenées à procéder à un rapprochement interdisciplinaire, dont le Traitement Automatique des Langues peut être le catalyseur, en tant que pourvoyeur de méthodes et outils de construction de RTO à partir de textes.

En terminologie, au cours des années 80, un rapprochement avec l'informatique s'est opéré avec le développement de la microinformatique. On s'est intéressé à la conception de bases de données terminologiques susceptibles d'aider les traducteurs professionnels dans les tâches de gestion et d'exploitation de lexiques multilingues. Les réflexions ont porté essentiellement sur le format de la fiche terminologique : à l'aide de quels champs décrire un terme dans une base de données qui sera utilisée par un traducteur humain ? Depuis la fin des années 90, la terminologie classique voit les bases théoriques de sa doctrine ainsi que ses rapports avec l'informatique ébranlés par le renouvellement de la pratique terminologique que suscite le développement des nouvelles applications de la terminologie. La multiplication des types de ressources terminologiques met à mal le principe théorique de l'unicité et de la fixité d'une terminologie pour un domaine donné, ainsi que celui de la base de donnée terminologique comme seul type de ressource informatique pour la terminologie. Depuis le milieu des années 90, un courant de recherche se développe autour de la terminologie textuelle, qui préconise la

construction de terminologies à partir de textes, et qui sollicite le TAL pour des méthodes et outils d'analyse de corpus (Slodzian, 2000). En Intelligence Artificielle, une évolution importante du domaine s'est produite de façon concomitante et parallèle à ce renouvellement théorique et méthodologique en terminologie. L'échec relatif des réalisations en IA a conduit à remettre en cause l'hypothèse qui était à la base du développement des systèmes experts, selon laquelle l'expert d'un domaine serait le dépositaire d'un système conceptuel qu'il suffirait de mettre au jour, en interrogeant l'expert ou en l'observant au travail. L'Ingénierie des Connaissances (IC) s'est alors imposée comme une direction de recherche en IA, avec pour ambition de résoudre les difficultés soulevées par la construction des systèmes experts, et de proposer des concepts, méthodes et techniques permettant d'acquérir et de modéliser les connaissances dans des domaines se formalisant peu ou pas. L'IC s'intéresse en particulier au processus de construction d'ontologies formelles pour les systèmes à base de connaissances ou pour l'interopérabilité entre systèmes dans le Web sémantique. Elle préconise elle aussi que, dans certains contextes, ce processus s'appuie sur l'analyse de corpus de textes. Elle sollicite le TAL pour des outils rendant possible et efficace la tâche de construction d'ontologies à partir de textes.

Des sollicitations analogues émanent aussi d'autres disciplines, comme les sciences de l'information et de la documentation. Au sein même du domaine du Traitement Automatique des Langues, certaines applications, comme la traduction automatique, la recherche d'information ou l'extraction d'information, ont besoin de ressources termino-ontologiques. Le TAL est donc ainsi doublement concerné par la problématique de la construction de RTO à partir de textes, en tant que consommateur de ressources et en tant que pourvoyeur d'outils pour les construire. Le TAL se trouve donc à la convergence de demandes émanant de disciplines diverses et concernant la mise à disposition d'outils et de méthodes d'analyse de textes pour la construction de ressources termino-ontologiques. Il peut adopter ainsi une position décalée par rapport à chacune de ces disciplines et saisir, grâce à cet angle de vue privilégié, les proximités et les différences entre des différents types de ressources, avec une objectivité et un recul, que ne peuvent avoir ces disciplines seules. En ce sens, le TAL peut favoriser le décloisonnement de ces disciplines et encourager le rapprochement pluridisciplinaire, autour d'une réflexion sur la notion de ressource termino-ontologique. Cette réflexion doit permettre de mettre en évidence les ressemblances et les particularités de ces différents types de ressources, de façon à spécifier les types d'outils d'analyse relativement génériques et utilisables pour une large gamme de ressources et de contextes d'exploitation.

2.2 Ontologie, terminologie, thesaurus, ...

Le TAL se trouve donc face à des disciplines chacune préoccupée par le problème de la construction de ressources termino-ontologiques de types différents, puisque destinées à des usages différents. Dans ce contexte de sollicitations diversifiées, il est non pertinent pour le TAL de se lancer dans une réflexion théorique visant à caractériser formellement et de façon générique ce qu'est une ressource termino-ontologique. L'approche consiste plutôt à mettre en perspective les différentes définitions travaillées par ces disciplines. L'objectif est de saisir en quoi les caractéristiques spécifiques de ces différents types de ressources dépendent des contextes applicatifs, pour finalement identifier ce qui différencie et, surtout, ce qui rapproche ces différents types de ressources. Il est alors possible de spécifier les différents types d'outils génériques de TAL dont il convient de promouvoir le développement.

Les réflexions sur les ontologies se sont d'abord développées en informatique (intelligence artificielle, sciences de la gestion), dans le cadre de travaux qui avaient comme objectif final la spécification de systèmes informatiques, avec plus particulièrement à l'origine la volonté de pouvoir réutiliser des composants génériques d'une application à une autre, ou encore de favoriser la communication entre différentes applications. C'est le cas encore des travaux menés en Ingénierie des Connaissances ou en représentation des connaissances autour des Systèmes à Base de Connaissances et du Web sémantique. Dans ce contexte, une ontologie est une conceptualisation des objets du domaine selon un certain point de vue, imposé par l'application. Elle est conçue comme un ensemble de concepts, organisés à l'aide de relations structurantes, dont la principale, celle avec laquelle est construite l'ossature de l'ontologie, est la relation *is-a*. Cette conceptualisation est écrite dans un langage de représentation des connaissances, qui propose des « services inférentiels » (classification de concept, capacité de construire des concepts définis à partir de concepts primitifs, etc.). A l'opposé, pour les thesaurus, un haut degré de formalisation et des services d'inférence ne sont pas nécessaires. Les thesaurus sont organisés avec les classiques relations d'hyperonymie et de synonymie, auxquelles s'ajoute la relation *voir aussi*. Néanmoins, il faut bien distinguer les thesaurus selon qu'ils sont exploités par des indexeurs et documentalistes humains, ou par des systèmes informatiques. Au cours d'une tâche d'indexation, pour choisir les meilleurs descripteurs, les agents humains procèdent à des interprétations et des inférences, qui s'appuient sur leur connaissance du domaine et des utilisateurs, connaissances implicites qui ne sont pas consignées dans le thesaurus. Les systèmes d'indexation automatique ne peuvent approcher de tels comportements intelligents qu'à condition que ces connaissances soient autant que possible explicitées et représentées dans les thesaurus, qui tendent ainsi à se rapprocher des ontologies de l'Ingénierie des Connaissances.

Le principal critère de discrimination entre RTO est le type de données d'entrée du système de traitement de l'information qui exploite la RTO. Selon que ces systèmes traitent de l'information de nature textuelle ou non, les caractéristiques des RTO vont être relativement différentes. Si le système analyse des entrée en langue naturelle, la première exigence est qu'il soit capable de reconnaître sous des formes linguistiques différentes des occurrences de la même unité et, inversement, de reconnaître des unités différentes sous une même forme. Il doit pouvoir gérer, aussi bien que l'application l'exige, les phénomènes de synonymie, de paraphrase, de variabilité linguistique aux niveaux morphologique ou syntaxique ou lexical, présents en masse dans les textes en langues naturelles (Zweigenbaum, 1999). Ceci n'est possible que si des règles de correspondance sont répertoriées dans la RTO que va exploiter le système. Une des tâches de l'analyste qui construit la RTO est donc de décrire des liens entre des motifs textuels et des unités de traitement, unités qui seront ensuite exploitées pour effectuer les traitements assignés aux système (classification de document, expansion de requête, extraction d'information, etc.). Quand les motifs textuels ont la structure de noms ou syntagmes nominaux, ils sont naturellement désignés sous le nom de termes. Les unités de traitement sont les concepts. C'est la raison pour laquelle nous parlons de *ressources termino-ontologiques*. De ce point vue, le concept peut être vu comme une classe d'équivalence de termes, ou plus généralement de motifs textuels, modulo les contraintes de l'application cible : deux motifs sont jugés équivalents, ou synonymes, en fonction de traitement que doit effectué par le système. Le concept est un mode de regroupement de termes. Ceci n'est pas incompatible avec sa fonction de regroupement d'objets (informatiques) du domaine qui lui est assignée dans les ontologies de l'Ingénierie des Connaissances. Le système de traitement de l'information dispose donc pour traiter de la synonymie de règles d'appariement qui

exploitent les liens termes/concepts présents dans la RTO. Il dispose de règles analogues pour le traitement de la polysémie, de l'homographie.

Si l'application cible n'est pas une application textuelle, l'analyse des textes n'en est pas moins fondamentale. Même s'il s'agit de construire une ontologie pour un système informatique, dont les données d'entrée ne seront pas textuelles, mais numériques, par exemple des résultats de mesures de capteur, l'analyse de textes et la description du vocabulaire sont néanmoins primordiales pour la construction de l'ontologie. En effet, l'analyse des textes sert d'indicateur à l'organisation d'un système conceptuel et donc à la mise en relation de concepts, et, par ailleurs, le choix des étiquettes de concepts doit être judicieux pour assurer l'interprétabilité et l'intelligibilité du système, ainsi que la maintenance de l'ontologie (Bachimont, 2000).

Cette position constructiviste et fonctionnelle des notions de terme et de concept s'éloigne quelque peu des positions référentialistes et fixistes - le terme comme étiquette de concept -, qui sont classiquement adoptées dans les domaines de l'Intelligence Artificielle, de la terminologie ou du Traitement Automatique des Langues, disciplines qui ont longtemps été largement influencées par une sémiotique du signe fondée sur la triade terme/concept/référent (Rastier, 1991). La conception classique pose que le terme existe en tant que représentant linguistique d'un concept faisant partie d'un système conceptuel unique et stable caractérisant a priori le domaine. Mais le constat de la variabilité des terminologies s'impose : étant donné un domaine d'activité, il n'y a pas *une* terminologie, qui représenterait le savoir sur le domaine, mais autant de ressources termino-ontologiques que d'applications dans lesquelles ces ressources sont utilisées. L'ensemble de ces constats empiriques appelle à un renouvellement théorique de la terminologie (Rastier, 1995) (Slodzian, 2000). A rebours de la conception fixiste et apriorique, on peut voir le terme et le concept comme le *résultat* d'un processus d'analyse termino-conceptuelle. Un mot ou une unité complexe n'acquiert le statut de terme que par décision. Dans le cas qui nous concerne ici, cette décision est prise par l'analyste en charge de l'élaboration d'une RTO pour une application bien identifiée. Celui-ci définit son propre référentiel de décision. Il procède à un travail de *construction* d'*une* ressource termino-ontologique pour une application dans le domaine, et non de *découverte* de la terminologie du domaine. Ce travail est guidé par une double contrainte de pertinence :

- pertinence vis-à-vis du corpus. Il s'agit de retenir et de décrire des structures lexicales qui présentent des caractéristiques à la fois spécifiques au domaine et stables dans le corpus ;
- pertinence vis-à-vis de l'application visée. Les unités finalement retenues doivent l'être en fonction de leur utilité dans l'application visée, qui s'exprime en termes d'économie, de cohérence interne et d'efficacité.

3 Eléments méthodologiques

3.1 Des outils d'aide

Devant la masse des données à analyser et étant donnés les délais de réalisation imposés, les disciplines concernées par la construction de RTO se tournent vers le TAL pour des outils informatiques d'analyse de corpus.

Les travaux de conception d'outils de TAL pour la construction de RTO doivent développer une réflexion méthodologique sur l'activité de construction elle-même. Doit s'imposer d'emblée le postulat que cette activité est avant tout une activité humaine, intellectuelle, menée par un individu que nous nommerons ici « analyste ». Dans un projet de construction de RTO, les contraintes sont multiples et multiformes, les choix à effectuer nombreux et de types divers, et ces choix comme ces contraintes, de type heuristique, sont difficilement explicitables. ont jusqu'ici été peu explicités. Par conséquent cette tâche ne peut en rien se limiter à l'élaboration automatique d'un réseau de termes et de concepts par quelque outil que ce soit. Nous défendons que la contribution du TAL doit être la fourniture d'outils d'aide pour l'analyste. Les recherches doivent se développer dans le paradigme de la coopération, et non celui de l'automatisation, même partielle, et il faut assumer, dans une perspective ingénierique, le rôle central de l'analyste. Autant les outils de TAL consommateurs de ressources termino-ontologiques doivent et peuvent approcher l'automaticité, autant les outils de TAL d'aide à la construction de RTO exigent l'intervention d'un agent humain.

Au-delà des difficultés techniques traditionnellement liées au développement d'outils en TAL, il existe une tension particulière propre au développement d'outils d'aide à la construction de RTO : il s'agit de concilier le caractère ad hoc des ressources à construire avec les outils, avec les contraintes de générnicité, transportabilité, reproductibilité, qu'impose le développement de la recherche. Autrement dit, il faut chercher à développer des outils de TAL relativement génériques quant au domaine et au type d'application, pour des utilisations elles très ciblées quant à ces deux points.

Rapidement, on peut classer les types d'outils à construire selon deux axes. Du point de vue fonctionnel, on peut distinguer les outils d'aide à l'acquisition de termes et les outils d'aide à la structuration de termes et au regroupement conceptuel (section 4). Du point de vue du mode d'utilisation, on peut distinguer les outils qui fonctionnent « en batch » (ils traitent l'ensemble du corpus, puis fournissent les résultats à l'analyste), et les outils interactifs. Par ailleurs, puisque les décisions prises par l'expert s'appuient *in fine* sur l'analyse de contextes dans le corpus, à côté des outils de traitement massif de corpus, il faut fournir à l'analyste des moyens d'accès au texte (concordanciers, outils de navigation hypertextuelle, etc.).

3.2 Rôle de l'analyste

Dans l'idéal, la personne chargée de construire la RTO, l'analyste, devrait avoir à la fois des compétences métier, des compétences en modélisation des connaissances et en linguistique et des compétences en informatique. Ce profil fait-il de l'analyste un oiseau rare ? Dans la réalité, il faut mettre en place une collaboration entre acteurs de spécialités différentes. Plusieurs sortes de situations peuvent être rencontrées. Pour les applications à forte dimension

cognitive, l'expérience montre que l'efficacité maximale peut être atteinte quand la construction de la RTO est assurée par un spécialiste métier, passionné par les problèmes de langue et de connaissance, ou formé à ceux-ci, qui comprend bien les spécifications de l'application cible et qui est capable de dialoguer avec les informaticiens qui la développent. A l'opposé, certaines applications, de type documentaire, ne requièrent pas une implication forte des spécialistes et la construction de la RTO peut être réalisée par des personnes ayant le profil et l'expérience de documentaliste ou de terminologue. Dans tous les cas, l'intervention d'un analyste médiateur est nécessaire quand l'application exige la participation de plusieurs spécialistes.

3.3 Place du corpus

Dans un projet de construction de RTO à partir de textes, la tâche de construction du corpus est à la fois primordiale et délicate. Puisque, d'une part, le corpus est la source d'information essentielle pour tout le processus de construction de la RTO et que, d'autre part, il restera, une fois le processus achevé, l'élément de documentation de la ressource construite, il doit être composé avec un maximum de précautions méthodologiques. Dans ce domaine, il n'est hélas pas encore possible de définir *a priori* des instructions méthodologiques très précises pour encadrer la tâche de sélection des sources textuelles qui viendront constituer le corpus. Au-delà des problèmes techniques ou politiques de disponibilité des textes, cette collecte doit se faire avec l'aide des spécialistes et en fonction de l'application cible visée. Il convient en effet de s'assurer auprès des spécialistes que les textes choisis ont un statut suffisamment consensuel pour éviter toute remise en cause ultérieure de la part d'utilisateurs ou de leur part. Par ailleurs, il convient de prévoir d'emblée une boucle de rétroaction au cours de laquelle une première version du corpus sera modifiée et enrichie en fonction d'une première phase d'analyse des résultats fournis par les outils de TAL sur cette version initiale. Le critère de la taille est évidemment important, même s'il est impossible de donner un chiffre idéal. Le choix est ici encore un compromis. Le corpus doit être suffisamment « gros » pour justifier que des outils de traitement de la langue soient nécessaires pour le dépouiller de façon efficace. Mais il doit être suffisamment petit et/ou redondant pour pouvoir être appréhendé de façon globale par l'analyste, même à l'aide d'outils de TAL. Une fourchette entre 50 000 et 200 000 mots semble raisonnable. Les projets prenant le Web comme source de textes font rapidement exploser ces chiffres, posant par la même des problèmes spécifiques, comme celui de la définition d'un « échantillon » pertinent pour l'étude. Enfin, dans la majorité des cas, le corpus sera hétérogène dans le sens où il aura été constitué en rassemblant des textes d'origine variée. Il est alors absolument nécessaire de procéder à un balisage du corpus qui permettra aux outils d'analyse, et ainsi qu'à l'analyste, de repérer les différents sous-corpus pour procéder éventuellement à des analyses contrastives.

3.4 Utilisation de ressources existantes

On l'aura compris, nous ne nous intéressons ici ni aux ontologies génériques (« à la Cyc ») censées représenter un ensemble maximal de connaissances, de sens commun, ni aux ontologies formelles (au sens de Guarino) qui constituerait un cadre référentiel universel et formellement valide, mais bien à des ressources termino-ontologiques exploitées par un système particulier de traitement de l'information dans un domaine particulier. C'est l'usage prévu de la ressource qui constraint et encadre sa construction. Pour autant, nous ne souhaitons

pas participer à une polémique sur l’opposition ontologies générales vs. ontologies spécialisées. Notre position est la suivante : il est primordial que les outils d’aide à la construction de RTO puissent recycler des données existantes afin de tirer le meilleur parti du patrimoine terminologique possédé par les entreprises et les institutions (Jacquemin, 1997). Pour une tâche de construction de RTO, il faut faire feu de tout bois, et chercher à exploiter autant que faire se peut toutes les ressources disponibles, et pas uniquement les textes. Sur le plan de la politique de la recherche, nous pensons qu’il est utile de promouvoir des travaux montrant l’utilité de ressources lexicales existantes (générales, comme la base WordNet ou des fichiers électroniques de synonymes, ou spécialisées, comme les grands thesaurus de la médecine comme UMLS) dans la perspective d’améliorer le rendement du couple analyste/outils de TAL. Nous sommes plus réservés sur la nécessité de dégager des financements lourds pour la réalisation de nouvelles ressources sémantico-conceptuelles de taille gigantesque, élaborées hors de toute spécification d’application cible. Il nous semble plus pertinent que soient encouragés des expériences d’évaluation, nécessairement très lourdes, proposant des protocoles expérimentaux capables de mettre en évidence à grande échelle les gains en temps et en qualité apportés par l’introduction d’une ontologie dans tel ou tel système de traitement de l’information par rapport au coût de la construction de cette ontologie (cf. section 6).

3.5 De la nécessité d’interface intégratrices

La tâche de construction d’une RTO est incrémentale et comporte de nombreux enchaînements d’essais/erreurs. Il faut des *interfaces* ergonomiques permettant une utilisation coordonnée et optimale des différents outils de traitement et de consultation du corpus de référence, par l’analyste qui construit une RTO, à l’instar de (Ait El Mekki et Nazarenko, 2002) pour la construction d’index d’ouvrages, de la plate-forme de modélisation TERMINAE pour la construction de terminologies et d’ontologies (Szulman et al., 2002). De façon plus générale, l’utilisation de ces différents outils doit être encadrée par une méthodologie précisant à quel stade du processus et selon quelles modalités il convient de les utiliser. En effet, la solution au problème de l’acquisition de ressources termino-ontologiques à partir de corpus ne réside pas uniquement en la fourniture d’un ou de plusieurs outils de traitement automatique des langues. La mise à disposition de tels outils doit s’accompagner d’une réflexion méthodologique poussée, conduisant à la réalisation de guides méthodologiques et de plates-formes logicielles intégratrices permettant la mise en œuvre efficace des outils proposés. Cette nécessité appelle une coopération entre TAL et IC. Cette réflexion sur l’utilisation combinée de différents types d’outils d’analyse de textes en ingénierie terminologique est aussi très présente dans un certain nombre de travaux en ingénierie des connaissances (Charlet et al., 2000).

3.6 Une proposition méthodologique

A titre d’exemple, nous évoquons une proposition méthodologique intégrant l’utilisation de plusieurs outils de TAL et qui se veut une réponse possible aux différents problèmes évoqués : la méthode TERMINAE (Szulman et al., 2002). Cette méthode s’appuie sur des travaux représentatifs du courant français de travaux à la convergence entre terminologie,

linguistique, ingénierie des connaissances et intelligence artificielle². Elle s'appuie sur les principes suivants :

- Partir de textes du domaine comme sources de connaissances : ils constituent un support tangible, rassemblant des connaissances stabilisées qui servent de référence et améliorent la qualité du modèle final ;
- Enrichir le modèle conceptuel d'une composante linguistique : l'accès aux termes et aux textes qui justifient la définition des concepts garantit une meilleure compréhension du modèle ;
- Utiliser des techniques et outils de TAL basés sur des travaux linguistiques : ils permettent l'exploitation systématique des textes et leurs résultats facilitent la modélisation ;
- Construire des ontologies « régionales », c'est-à-dire consensuelles dans un domaine et adaptées à une application, mais non universelles ;
- Appliquer des principes de modélisation systématiques pour assurer une bonne structuration des données et faciliter la maintenance de l'ontologie.

TERMINAE vise essentiellement la constitution de terminologies, réseaux conceptuels et ontologies. La méthode comprend quatre étapes, les trois dernières étant mises en oeuvre de manière cyclique. L'importance de chacune dépend du produit terminologique visé et des objectifs d'utilisation de ce dernier.

- La Constitution d'un corpus vise à choisir documents techniques, comptes rendus, livres de cours, etc. à partir d'une analyse des besoins de l'application.
- L'étude linguistique consiste à identifier des termes et des relations lexicales, en utilisant des outils de traitement de la langue naturelle (SYNTEX comme extracteur de termes, UPPERY comme outil d'analyse distributionnelle, Caméléon pour l'aide au repérage de relations par des patrons linguistiques, YAKWA comme concordancier).
- La normalisation sémantique conduit à définir dans un langage formel des concepts et des relations sémantiques que nous appelons terminologiques car provenant des termes et relations précédemment étudiés (Biébow & Szulman, 1999). Leur structuration en réseau s'appuie sur les résultats du dépouillement des textes tout en tenant compte de l'objectif d'utilisation de l'ontologie. Elle nécessite l'ajout de nouveaux concepts et relations dits de structuration.
- La formalisation permet de préciser, compléter et valider le modèle construit lors de la nor-malisation. L'analyste indique si les concepts sont primitifs ou définis, vérifie que les relations sont à la bonne place pour favoriser un héritage maximum, etc.

Le logiciel TERMINAE associé à la méthode fournit des aides pour toutes les étapes de l'analyse des textes à la formalisation. Il offre un support méthodologique qui permet d'évoluer progressivement et en conservant des liens des textes vers les niveaux linguistique

² Ce courant, animé au sein du GDR-I3 et de l'AFIA par le groupe TIA (<http://www.biomath.jussieu.fr/TIA/>) dont les auteurs font partie.

et conceptuel. Le logiciel assure donc une continuité entre les différentes formes de l'ontologie. Celle-ci passe d'un état proche d'une taxinomie de termes à un réseau conceptuel enrichi de relations et de concepts de structuration pour aboutir à une ontologie formelle. Elle est décrite dans un langage formel masqué à l'analyste qui permet de vérifier des contraintes de validité minimale.

4 Outils de TAL pour la construction de RTO

4.1 Une typologie fonctionnelle

Dans cette section³, nous passons en revue un certain nombre de travaux de recherche sur le développement d'outils d'aide à la construction de RTO à partir de textes. Nous avons choisi de les présenter selon une typologie de fonctionnelle.

- *Acquisition de termes.* Une première classe regroupe les outils dont la visée est l'extraction à partir du corpus analysé de *candidats termes*, c'est-à-dire de mots ou groupes de mots susceptibles d'être retenus comme termes par un analyste, et de fournir des étiquettes de concepts. Ces outils diffèrent principalement quant au type de techniques mises en œuvre (syntaxique, statistique, autres).
- *Structuration de termes et regroupement conceptuel.* Les ressources termino-ontologiques se présentent rarement sous la forme d'une liste à plat. Des outils d'aide à la structuration d'ensembles de termes sont donc nécessaires. Dans cette classe, nous évoquerons, d'une part, des outils de classification automatique de termes, et, d'autre part, des outils de repérage de relation. Signalons que beaucoup d'outils d'extraction proposent déjà une structuration des candidats termes extraits.

4.2 Acquisition de termes

L'outil TERMINO est une application pionnière de l'acquisition automatique de termes (David et Plante, 1990). Construit sur la base de l'atelier FX, un formalisme pour l'expression de grammaires du langage naturel et un analyseur associé, TERMINO se focalise sur le repérage des syntagmes nominaux qui sont les seules structures supposées produire des termes. Les candidats termes extraits par TERMINO sont appelés "synapsies" d'après les travaux de Benveniste. La chaîne de traitement de TERMINO se compose d'une phase d'analyse morphosyntaxique suivie d'une phase de génération des synapsies à partir des dépendances entre tête et compléments rencontrés dans la structure de syntagme nominal retournée par l'analyseur. ANA est un outil d'acquisition terminologique qui extrait des candidats termes sans effectuer d'analyse linguistique (Enguehard et Pantera, 1995). Les termes sont reconnus au moyen d'égalités approximatives entre mots et d'une observation de répétitions de patrons. ACABIT extrait des candidats termes à partir d'un corpus préalablement étiqueté et désambiguïsé (Daille, 1994). ACABIT mêle des traitements linguistiques et des filtres

³ Cette partie est extraite et adaptée de (Bourigault et Jacquemin, 2000) parue dans l'ouvrage *Industrie des langues* (Hermès) coordonné par J.-M. Pierrel. Une bibliographie mise à jour sera fournie lors du cours.

statistiques. L'acquisition terminologique dans ACABIT se déroule en deux étapes : (1) analyse linguistique et regroupement de variantes, au cours de laquelle n ensemble de transducteurs analyse le corpus étiqueté pour extraire des séquences nominales et les ramener à des candidats termes binaires ; (2) filtrage statistique, au cours duquel les candidats termes binaires produits à l'étape précédente sont triés au moyen de mesures statistiques. A l'instar d'ACABIT, LEXTER extrait des candidats termes à partir d'un corpus préalablement étiqueté et désambiguïsé (Bourigault, 1994). Il effectue une analyse syntaxique de surface pour repérer les syntagmes nominaux maximaux, puis une analyse syntaxique profonde pour analyser et décomposer ces syntagmes. Il est doté de procédures d'apprentissage endogène pour acquérir des informations de sous-catégorisation des noms et adjectifs propres aux corpus. Il organise l'ensemble des candidats termes extraits sous la forme d'un réseau. FASTR est un analyseur syntaxique robuste dédié à la reconnaissance en corpus de termes appartenant à une liste contrôlée fournie au système (Jacquemin, 1997). Les termes n'ayant pas toujours, en corpus, la même forme linguistique, le principal enjeu est de pouvoir identifier leurs variantes. FASTR est doté d'un ensemble élaboré de métarègles, qui lui permettent de repérer différents types de variation : les variantes syntaxiques, morpho-syntaxiques et sémantico-syntaxiques. L'environnement SYMONTOS (Velardi et al., 2001) propose des outils pour repérer des termes simples et complexes dans des textes et des critères pour décider de définir des concepts à partir de ces termes.

4.3 Structuration de termes et regroupement conceptuel

La gamme des outils d'aide à la structuration de terminologie est très large. Sont susceptibles d'émerger à cette catégorie un certain nombre de types d'outils qui n'étaient pas initialement conçus spécifiquement pour cette tâche, mais qui ont été développés pour des applications d'informatique documentaire ou d'extraction d'informations, par exemple. Nous balayons rapidement un spectre assez large, couvrant les outils de classification de termes sur la base de cooccurrences dans des textes ou dans des fenêtres, les outils de classification de termes sur la base de distributions syntaxiques et les outils de repérage de relations. Les outils de cooccurrence développés dans le domaine de la recherche d'information rapprochent des termes qui apparaissent fréquemment dans les mêmes (portions de) documents, et qui possèdent donc sans doute une certaine proximité sémantique. La technique de recherche de cooccurrences est déjà ancienne (à l'échelle de l'histoire de l'informatique) puisqu'elle a été promue très tôt en informatique documentaire pour permettre l'expansion de requêtes (Sparck Jones, 1971). Parmi les applications dans le domaine de l'acquisition terminologique, on peut citer le projet ILIAD (Toussaint et al., 1998), et les travaux de G. Lame (2002). Toujours dans le domaine de l'informatique documentaire, les travaux dans le domaine de la construction automatique de thesaurus peuvent être réinvestis dans des applications terminologiques. Par exemple, la chaîne de traitement développée par G. Greffenstette construit automatiquement des classes comportant des noms qui se retrouvent régulièrement comme arguments des mêmes verbes (Greffenstette, 1994). Ce repérage de la position argumentale des noms se fait grâce à l'exploitation d'un analyseur syntaxique de surface à large couverture. Ces techniques inspirées de la linguistique harrissienne, qui visent à rapprocher les termes qui ont des distributions syntaxiques analogues, sont à la base de nombreux travaux depuis plusieurs années (Assadi, 1998) (Habert et al, 1996) (Faure, 2000).

Les outils que nous venons d'évoquer visent à rapprocher des termes à partir d'une analyse globale de l'ensemble de leurs occurrences. Ils ne touchent que les termes fréquents, et donc

le plus souvent des noms simples, et proposent une simple relation d'équivalence (appartenance à une classe). A côté de ces outils qui travaillent sur les types comme regroupement des occurrences, on trouve les outils de repérage de relations, qui travaillent au niveau des occurrences elles-mêmes. Ces outils détectent en corpus des mots ou contextes syntaxiques répertoriés comme susceptibles de “ marquer ” telle ou telle relation entre deux éléments. Les travaux de M. Hearst, sur l'extraction automatique des liens d'hyperonymie, font figure de référence (Hearst, 1992). Les recherches sur ce thème se déclinent de multiples façons. L'un des enjeux principaux concerne la généralité des relations, et celles des marqueurs de relations. D'un côté, il existe probablement des relations que l'on jugera toujours pertinentes pour décrire un domaine de connaissance, par exemple les relations de type hiérarchique ou partitive, et des marqueurs pour ces relations eux aussi généraux (Garcia, 1998). A l'opposé, il est indéniable que chaque domaine est structuré par des relations qui lui sont spécifiques, et qu'il convient nécessairement de prendre en compte pour décrire le domaine. De plus même dans le cas de relations considérées comme générales, il est possible que les marqueurs susceptibles de conduire à les identifier diffèrent d'un corpus à l'autre. Se pose alors le problème de l'apprentissage inductif de ces marqueurs de relation. Un certain nombre de travaux en TAL et en IC sont consacrés à ce problème. Ils partent tous du même principe d'une recherche itérative alternée dans le corpus à la fois des marqueurs d'une relation donnée et des couples de termes qui entrent dans cette relation (Rousselot et al., 1996) (Séguela et Aussenac-Gilles, 1999) (Morin, 1999) (Condamines et Rebeyrolles, 2000) (Maedche et Staab, 2000).

5 Trois retours d'expérience

5.1 Contextes

Les exemples, démonstrations et expérimentations proposés pendant le cours sont issus principalement de 3 expériences réelles de construction de RTO à partir de textes⁴. Ces trois expériences couvrent un spectre large de types de domaines et de types d'applications : la première expérience a été menée dans le domaine technique de la fabrication du verre (projet VERRE), avec comme application cible la classification de documents ; la deuxième expérience a été menée dans un domaine médical de la réanimation chirurgicale (projet REA), avec comme application cible le codage d'actes médicaux ; la troisième expérience a été menée dans le domaine juridique du Droit français codifié (projet DROIT), avec comme application cible l'aide à la reformulation de requêtes. Il s'agit à chaque fois de projets de Recherche et Développement, dans lesquels l'application cible n'est pas strictement spécifiée au départ du projet, comme cela devrait l'être dans un « vrai » projet industriel. On doit donc être prudent au moment de tirer des conclusions générales. Néanmoins, chacun de ces projets est allé à son terme, en ce sens qu'il n'a pas conduit à des RTO « jouets », mais à des ressources complètes qui sont ou seraient exploitables. Par ailleurs, chaque projet a permis de tester certaines hypothèses méthodologiques faisant ainsi progresser les recherches dans le domaine de l'acquisition des connaissances à partir de textes. C'est en multipliant ce type

⁴ Cette partie est extraite et adaptée d'un article à paraître dans un numéro spécial de la Revue d'Intelligence Artificielle, coordonné par M. Slodzian et J.-M. Pierrel (Aussenac-Gilles & al, 2003)

d'expériences que l'on avancera sur la définition d'un cadre méthodologique relativement précis qui aille au-delà d'un simple recueil de bonnes pratiques et qui puisse satisfaire les exigences d'un transfert vers les applications industrielles.

5.1.1 *Le projet VERRE : une ontologie dans le domaine de la fabrication et d'utilisation de la fibre de verre*

Le premier projet vient répondre à une demande du centre de recherche du groupe Saint-Gobain. Au sein des différentes filiales du groupe, l'avance technologique et industrielle est primordiale pour conserver une place compétitive par rapport aux entreprises concurrentes. Les activités de veille documentaire et technologique jouent alors un rôle crucial, et font l'objet d'un outillage informatique de plus en plus performant. Parmi ces activités, une demande récurrente des documentalistes porte sur la définition d'un outil d'aide au repérage de nouveaux documents pertinents sur le Web (comme des brevets, des dépêches de presse, etc.) et à leur classement en fonction des domaines d'intérêt des ingénieurs qui les consultent. Or la plupart des outils de routage de documents s'appuient sur un réseau conceptuel d'autant plus performant qu'il est enrichi des connaissances et de la terminologie du domaine de l'entreprise. L'objectif du projet était donc de tester la faisabilité du développement d'une ontologie dans l'objectif de l'utiliser pour guider le classement de documents en fonction des profils des utilisateurs. Dans ce projet, les aspects méthodologiques étaient tout aussi importants que l'ontologie elle-même. L'étude a été menée par deux chercheurs de l'IRIT, A. Busnel pour l'analyse terminologique et ontologique, et N. Aussenac-Gilles sur les aspects méthodologiques. Un début d'ontologie (50 concepts, 20 relations) a été mis en forme à l'aide du logiciel de modélisation TERMINAE, à partir de l'analyse d'un corpus de langue anglaise composé de différents types de documents sur le domaine. Les logiciels de Traitement Automatique des Langues SYNTTEX, UPERY et YAKWA ont été utilisés pour le dépouillement de ces corpus. Une proposition méthodologique utilisable dans le contexte de cette entreprise et pour ce type d'application a été mise en forme (Aussenac-Gilles & Busnel, 2002).

5.1.2 *Le projet REA : une ontologie dans le domaine de la traumatologie en réanimation chirurgicale*

Le deuxième projet a été encadré par M.-C. Jaulent et J. Charlet et a été mené à bien au sein de l'UFR Broussais-Hôtel-Dieu. Le contexte est celui du codage des actes médicaux par les médecins. Pour leur activité de codage obligatoire, les praticiens s'aident d'un thésaurus de spécialité qui a été élaboré de façon à ce que les séjours de réanimation soient le mieux possible valorisés. Il est aujourd'hui reconnu que l'ambiguïté du thésaurus est une source d'erreurs et de disparités de codage. Dans un domaine particulier tel que la réanimation chirurgicale, on ne peut envisager de réaliser des outils informatiques d'aide au codage qu'après avoir préalablement organisé des objets du domaine, en fonction de la tâche à résoudre, par le biais d'une ontologie. L'objectif de ce deuxième projet était donc de construire une ontologie du domaine de la réanimation chirurgicale. Les outils de Traitement Automatique des Langues SYNTTEX et UPERY ont été utilisés pour traiter un corpus de comptes rendus d'hospitalisation. Le travail a été réalisé par S. Le Moigno, médecin spécialiste, dans le cadre d'un stage de DEA en informatique médicale. L'ontologie comprend environ 2 000 concepts et 200 liens (Le Moigno et al., 2002).

5.1.3 Le projet DROIT : une ressource ontologique dans le domaine du Droit

Le troisième projet a été mené par G. Lame, au cours de sa thèse au Centre de Recherche en Informatique de l'Ecole des Mines de Paris (Lame, 2002). Ce centre de recherche a créé et héberge le site juridique droit.org, qui diffuse l'édition *Lois et décrets* du Journal Officiel de la République française, ce qui représente 95 000 documents (lois, décrets, arrêtés), ainsi que les codes du droit français (Code civil, Code pénal, etc.) et des textes européens (directives, règlements). L'objectif du travail était de tester l'intérêt et la faisabilité d'une approche consistant à intégrer une ontologie du Droit susceptible de faciliter l'accès au site par les utilisateurs. Le résultat est une ressource ontologique de très large couverture, couvrant tous les domaines du Droit, constituée d'environ 130 000 termes et 200 000 liens. Cette ressource est utilisée comme support pour un système d'expansion de requêtes : à un mot posé par l'utilisateur, le système propose tous les termes reliés à ce mot dans la ressource et laisse l'utilisateur choisir ceux qu'ils souhaitent retenir pour modifier sa requête. Cette ressource a été construite en utilisant les résultats bruts, sans aucun filtrage manuel, de différents outils ou techniques de Traitement Automatique des Langues (SYNTEX, cooccurrence statistique, UPERY), obtenus par analyse d'un corpus constitué de l'ensemble des Codes de la législation française.

5.2 Trois outils de TAL pour la construction de RTO à partir de textes

5.2.1 Extraction de termes : SYNTEX

Dans les trois projets, les résultats de l'outil SYNTEX ont été utilisés. SYNTEX (Bourigault et Fabre, 2000) est un analyseur syntaxique de corpus. Il existe actuellement une version pour le français, qui a été utilisée dans les projets REA et DROIT, et une version pour l'anglais, qui a été utilisée dans le projet VERRE. Après l'analyse syntaxique en dépendance de chacune des phrases du corpus, SYNTEX construit un réseau de mots et de syntagmes (verbaux, nominaux, adjetivaux), dit « réseau terminologique », dans lequel chaque syntagme est relié d'une part à sa tête et d'autre part à ses expansions. Les éléments du réseau (mots et syntagmes) sont appelés « candidats termes ».

A chaque candidat terme sont associées un certain nombre d'informations numériques, sur lesquelles l'utilisateur peut se baser pour organiser son dépouillement :

- *fréquence* : c'est le nombre d'occurrences du candidat terme détectées par le logiciel dans le corpus. L'interface d'analyse des résultats permet à l'analyste d'accéder à l'ensemble des contextes d'apparition du candidat terme dans le corpus. Cet accès au texte est d'autant plus crucial que l'utilisateur n'est pas un spécialiste du domaine.
- *productivité en Tête (resp. Expansion)* : c'est le nombre de « descendants en Tête » (resp. « descendants en Expansion ») du candidat terme, c'est-à-dire le nombre de candidats termes plus complexes qui ont le candidat terme en position tête (resp. expansion). A partir de ces informations, l'analyste peut visualiser des listes paradigmatisques de candidats termes partageant la même tête ou la même expansion (cf. figure 1), ce qui le guide vers la constitution de taxinomies locales.

La difficulté essentielle pour l'utilisateur vient de la masse des résultats fournis par l'extraction. Même s'il existe de nombreux travaux fort intéressants sur le filtrage statistique de candidats termes extraits automatiquement de corpus, l'expérience montre qu'aucune mesure statistique ne peut suppléer l'expertise de l'analyste, en particulier parce qu'il y a toujours des candidats termes de fréquence 1 dont l'analyse est intéressante. De façon générale, sachant qu'il ne pourra analyser tous les candidats termes extraits du corpus, l'analyste doit adopter une stratégie optimale qui, étant donné le temps qu'il a choisi de consacrer à la tâche d'analyse textuelle et en fonction du type de la ressource à construire, lui garantit que, parmi les candidats qui auront échappé à son analyse, la proportion de ceux qui auraient pu être pertinents est faible.

5.2.2 Analyse distributionnelle : UPERY

Dans les trois projets, les résultats de l'outil UPERY ont été utilisés. UPERY (Bourigault, 2002) est outil d'analyse distributionnelle. Il exploite l'ensemble des données présentes dans le réseau de mots et syntagmes construits par SYNTTEX pour effectuer un calcul des proximités distributionnelles entre ces unités. Ce calcul s'effectue sur la base des contextes syntaxiques partagés. Il s'agit d'une mise en œuvre du principe de l'analyse distributionnelle du linguiste américain Z. S. Harris, réalisée dans la lignée des travaux de H. Assadi (Assadi & Bourigault, 1996). L'analyse distributionnelle rapproche d'abord deux à deux des candidats termes qui partagent un grand nombre de contextes syntaxiques. Par exemple, dans le corpus REA, les candidats termes *insuffisance rénale* et *détresse respiratoire* sont rapprochés car on les trouve dans les contextes syntaxiques suivants : complément de *prise en charge*, de *apparition*, de *installation*, de *admettre en réanimation chirurgicale pour*.

Trois mesures permettent d'appréhender la proximité entre deux candidats termes. Le coefficient *a* est égal au nombre de contextes syntaxiques partagés par les deux termes. Cette mesure donne une première indication de la proximité entre deux termes. Mais cette mesure reflète de façon insatisfaisante la proximité. Il faut tenir compte de la productivité en Tête des contextes partagés : plus un contexte partagé par deux termes est productif, moins sa contribution au rapprochement des deux candidats termes doit être importante. Cette intuition est prise en compte par le coefficient *prox* qui pondère chaque contexte partagé par l'inverse de sa productivité. Enfin, pour évaluer la proximité entre deux unités, il est important de tenir compte non seulement de ce qu'elles partagent, mais aussi de ce qu'elles ont en propre. On caractérise la proximité entre deux candidats termes à l'aide de deux indices supplémentaires : pour chacun des deux candidats termes, le rapport entre le nombre de contextes partagés et le nombre total de contextes dans lesquels apparaît le candidat terme.

Le module d'analyse distributionnelle UPERY calcule chacun de ces coefficients pour chaque couple de candidats termes, et ne sont présentés à l'utilisateur que les couples dont les coefficients dépassent certains seuils. Ceux-ci sont définis de façon empirique et varient en fonction d'une part de l'homogénéité et de la redondance du corpus et d'autre part du contexte dans lequel doivent être exploités les résultats de l'analyse distributionnelle. L'analyse distributionnelle implémentée dans UPERY est symétrique : on calcule aussi la proximité entre contextes syntaxiques. Deux contextes syntaxiques sont proches si on y trouve les mêmes termes. Par exemple, dans le corpus REA, les verbes *montrer* et *mettre en évidence* sont proches car ils partagent en position sujet les termes *échographie*, *bilan infectieux*, *tomodensitométrie*, *artériographie*, *auscultation pulmonaire*, etc.

Il s'avère que les rapprochements effectués par UPERY sont extrêmement utiles et pertinents pour la construction de classes conceptuelles. Le nombre de rapprochements effectués dépend de la redondance du corpus. Par exemple, les corpus REA et le corpus du Code civil, l'un des corpus exploité dans le projet DROIT, sont deux corpus différents quant à ce paramètre de la redondance. Le corpus REA est constitué dans un ensemble de comptes rendus médicaux qui décrivent tous les mêmes types d'événement et donc dans lesquels les mêmes structures syntaxiques reviennent régulièrement. A l'opposé, dans le Code civil, les redondances, répétitions, reformulations sont évitées. Cela se répercute de façon assez sensible sur la richesse des résultats fournis par UPERY sur chacun des 2 corpus, puisque dans le corpus REA 30% des syntagmes nominaux et 47% des noms, de fréquence supérieure ou égale à 5, sont rapprochés d'au moins un autre mot, alors qu'ils ne sont que respectivement 20% et 43% pour le corpus du Code civil. Le phénomène est encore plus accentué dans le corpus LIVRE du projet VERRE. La taille du corpus est relativement réduite (100 000 mots, contre 400 000 pour le corpus REA et 150 000 pour le Code civil) et les redondances sont très faibles (chaque chapitre traite d'un sujet spécifique, et l'auteur s'efforce de varier son style). De ce fait, seuls 3% des SN et 18% des noms, de fréquence supérieure ou égale à 5, ont des voisins.,.

5.2.3 Extraction des relations : YAKWA et CAMELEON

Développé à l'ERSS par L. Tanguy, YAKWA est un concordancier pour corpus étiquetés (Rebeyrolle et Tanguy, 2000). Il permet de rechercher des phrases et/ou des paragraphes contenant une séquence définie par des marqueurs. Ce marqueurs s'appuient sur les informations notées par l'étiqueteur dans les corpus, comme les catégories grammaticales des mots. Leur contenu peut être formé de formes lexicales (tronquées, exactes, etc.), de formes canoniques des unités lexicales du texte, de catégories morpho-syntactiques et de leur combinaisons (disjonctions, conjonction de marqueur lexical et de marqueur morpho-syntactique), de la négation d'un des types de marqueurs précédents ou de jokers (mots non comptabilisés). YAKWA peut s'adapter à tout type d'étiqueteur, par exemple CORDIAL université pour le français ou TREE TAGGER pour l'anglais. Son interface guide la construction de marqueurs et permet d'en visualiser la projection sur un corpus.

CAMELEON est un logiciel de recherche de relations lexicales à partir de marqueurs linguistiques (Séguéla, 1999). Il est associé à un module de modélisation qui permet de valider (ou de rejeter) ces relations lexicales pour les intégrer sous forme de relations sémantiques dans un modèle conceptuel. Les marqueurs utilisés dans CAMELEON peuvent être des marqueurs génériques prédéfinis ou leur adaptation ou encore des marqueurs spécifiques définis par l'utilisateur. L'idée est de rechercher des relations avec des moyens adaptés au corpus étudié. Les relations sont donc génériques (comme EST-UN) ou spécifiques au corpus (comme « used-in » dans le projet VERRE), et les marqueurs associés à toutes les relations sont revus et adaptés à chaque corpus. Le langage d'expression des marqueurs est moins riche que celui de YAKWA car CAMELEON fonctionne sur un corpus brut non étiqueté. En revanche, Caméléon présente deux points forts pour la construction de RTO : il propose une base générique de relations et de marqueurs associés ; il s'appuie sur les résultats de SYNTEX pour suggérer les concepts qui pourraient être en relation à partir de la forme lexicale trouvée.

6 Le problème de l'évaluation

Nous terminerons par quelques réflexions sur le problème de l'évaluation. Il faut distinguer l'évaluation d'une RTO particulière construite dans un contexte particulier, de l'évaluation de tel outil ou tel outil de TAL d'aide à la construction de RTO. Dans les deux cas, il faut adopter une approche ingénierique, en adoptant les principes de base du génie logiciel, ce qui exige, a minima, de prendre en compte autant que possible le contexte global d'utilisation de la RTO ou de l'outil.

En ce qui concerne les RTO, il faut distinguer *validation* et *évaluation*. Dans le processus de construction d'une RTO, il y a plusieurs moment de *validation* de la RTO, c'est-à-dire de moment où l'analyste présente la ressource à l'experts (ou à des experts), et lui (leur) demande de valider ou d'invalider certains choix de modélisation effectués. Ces moments de validation sont d'autant moins nombreux que les experts sont peu disponibles. Ce sont donc des étapes très importantes dans le processus. L'enjeu est de s'assurer avec les experts que la conceptualisation représentée dans la RTO n'est pas en contradiction sur tel ou tel point avec les connaissances expertes. Le problème ne se pose pas tant en terme de vérité, qu'en terme de non violation des connaissances de l'expert. En effet, pour construire la modélisation, l'analyste a adopté un point de vue, celui de l'application cible dans laquelle sera intégrée la ressource, qui n'est pas nécessairement exactement celui de l'expert dans son activité. La tâche n'est pas simple. L'analyste doit aider l'expert, qui ne reconnaît pas nécessairement à première vue ses petits, à prendre le recul nécessaire pour déceler la présence d'erreurs, voire d'absences, flagrantes. Une fois la RTO construite, s'engage un processus d'*évaluation*. Comme nous l'avons déjà évoqué, l'évaluation doit être réalisée selon les procédures de base du génie logiciel. Il s'agit de vérifier si la RTO satisfait bien le cahier des charges et répond aux attentes spécifiées au début du projet. La difficulté, habituelle, est que l'ontologie n'est qu'un élément de l'application cible, qui est le dispositif à valider. Il faut donc concevoir des expériences et des bancs d'essais qui permettent de cibler l'évaluation sur la seule ressource. Une fois ces généralités affirmées, nous pouvons difficilement aller au-delà, parce que nous manquons encore de retour d'expérience, et parce que chaque cas étant particulier il sera de toutes façons difficile de définir des procédures à la fois précises et relativement génériques, et que cela dépasse quelque peu le cadre de la recherche.

L'évaluation des outils de construction de RTO est le problème qui nous concerne ici. C'est un problème lui aussi difficile. La source des difficultés est double : d'abord il s'agit d'outils d'aide, ensuite chaque outil est rarement utilisé seul. Quand il s'agit d'évaluer d'un outil automatique, du type « boîte noire », il est possible d'évaluer les performances de l'outil en comparant les résultats qu'il fournit à des résultats attendus (« gold standard »). En revanche, la situation est plus complexe dans le cas des outils d'aide qui nous intéressent ici. Les résultats fournis par les outils sont interprétés par l'analyste, et le résultat de cette interprétation est variable : une modification, un enrichissement de la ressource à un ou plusieurs points du réseau, voire dans certain cas l'absence d'action immédiate, sans que cela signifie nécessairement que les résultats en question soient faux ni même pertinents. De plus, chaque cette interprétation s'appuie normalement sur une confirmation par retour aux textes. Il n'y a pas systématiquement de trace directe entre un résultats (ou un ensemble de résultats) de l'outil et telle ou telle portion de la ressource. Si on rajoute à cela, qu'une portion de RTO n'a de sens que dans la globalité de la ressource, et la ressource elle-même ne peut être évaluée qu'en contexte, on saisit l'ampleur de la tâche.. Il y a un tel parcours interprétatif entre les résultats de l'outil et la ressource construite que le mode d'évaluation par

comparaison entre les résultats de l'outil et une ressource de référence ne peut apporter limités, même si cela peut donner des indications très intéressantes pour faire évoluer l'outil (Nazarenko et al., 2001). Là encore, nous n'avons de solution miracle à proposer. L'idéal serait par exemple de comparer entre termes de temps de réalisation et de qualité deux ressources ontologiques, l'une construite avec tel outil, et l'autre sans. Quand on connaît le temps de développement d'une ontologie, on imagine la lourdeur, et la difficulté de mise en œuvre d'une telle méthodologie. Le problème reste ouvert. Pour mesurer, ne serait-ce que d'un point de vue qualitatif, l'intérêt des outils, considérons pour le moment qu'il est primordial de les tester dans des contextes nombreux et variés et aussi réels que possible pour faire avancer la recherche.

Référence

- Ait El Mekki T., Nazarenko A (2002), Comment aider un auteur à construire l'index d'un ouvrage ?, Actes du *Colloque International sur la Fouille de Texte CIFT'2002*, Y. Toussaint et C. Nedellec Eds., oct. 2002, pp. 141-158
- Assadi H. (1998), *Construction d'ontologies à partir de textes techniques. Application aux systèmes documentaires*, thèse de l'Université Paris 6
- Aussenac-Gilles N. (1999), GEDITERM, un logiciel de gestion de bases de connaissances terminologiques, in Actes des Journées Terminologie et Intelligence Artificielle (TIA'99), Nantes, *Terminologies Nouvelles* n°19, 111-123.
- Aussenac N., Séguéla P. (2000), Les relations sémantiques : du linguistique au formel. *Cahiers de grammaire*, N° spécial sur la linguistique de corpus. A. Condamines (Ed.) Vol 25. Déc. 2000. Toulouse : Presse de l'UTM. Pp 175-198.
- Aussenac-Gilles N., Biébow B., Szulman N. (2000), Revisiting Ontology Design: a method based on corpus analysis. *Knowledge engineering and knowledge management: methods, models and tools, Proc. of the 12th International Conference on Knowledge Engineering and Knowledge Management*. Juan-Les-Pins (F). Oct 2000. R Dieng and O. Corby (Eds). Lecture Notes in Artificial Intelligence Vol 1937. Berlin: Springer Verlag. pp. 172-188.
- Bachimont, B. (2000), Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances ». In J. Charlet et al. (eds), *Ingénierie des Connaissances ; Evolutions récentes et nouveaux défis*, Eyrolles, pp. 305-323
- Bourigault D. (2002), Upéry : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus, Actes de la 9^{ème} conférence annuelle sur le Traitement Automatique des Langues (TALN 2002), Nancy, 2002, pp. 75-84
- Bourigault D., Fabre C. (2000), Approche linguistique pour l'analyse syntaxique de corpus, *Cahiers de Grammaires*, n° 25, 2000, Université Toulouse - Le Mirail, pp. 131-151.
- Bourigault D. & Jacquemin C. (2000), Construction de ressources terminologiques, in J.-M. Pierrel (éd.), *Industrie des langues*, Hermès, Paris, pp. 215-233

Charlet J., Zacklad M., Kassel G. & Bourigault D. (eds) (2000), *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, Eyrolles : Paris - Collection technique et scientifique des télécommunications

Charlet J. (2002), *L'ingénierie des connaissances : résultats, développements et perspectives pour la gestion des connaissances médicales*. Mémoire d'habilitation à diriger des recherches, université Pierre et Marie Curie

Chaumier J. (1988), *Travail et méthodes du/de la documentaliste : Connaissance du problème, Applications pratiques*. 3^e éd. mise à jour et complétée. Paris : ESF, 1988

Condamin A. et Rebeyrolles J (2000), Construction d'une base de connaissances terminologiques à partir de textes : expérimentation et définition d'une méthode. In Charlet J, Zacklad M., Kassel G. & Bourigault D. éds. *Ingénierie des connaissances. Tendances actuelles et nouveaux défis*. Editions Eyrolles/France Telecom, Paris

Daille B. (1994), *Approche mixte pour l'extraction de terminologie : statistique lexicale et filtres linguistiques*. Thèse en Informatique Fondamentale, Université de Paris 7, Paris

David S. et Plante P. (1990), De la nécessité d'une approche morpho-syntaxique dans l'analyse de textes. *Intelligence Artificielle et Sciences Cognitives au Québec*, 3(3):140-154

Enguehard C. et Pantera L. (1995), Automatic natural acquisition of a terminology. *Journal of Quantitative Linguistics*, 2(1):27-32

Faure D. (2000), *Conception de méthode d'apprentissage symbolique et automatique pour l'acquisition de cadres de sous-catégorisation de verbes et de connaissances sémantiques à partir de textes : le système ASIUM*, thèse de Doctorat Université de Paris Sud

Garcia D. (1998), *Analyse automatique de textes pour l'organisation causale des actions. Réalisation du système informatique COATIS*. Thèse en informatique. Université Paris IV

Grefenstette G. (1994), *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publisher, Boston, MA

Habert B., Naulleau E. et Nazarenko A . (1996), Symbolic word clustering for medium-size corpora. *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, Copenhagen, pp 490-495

Jacquemin C. (1997), *Variation terminologique : Reconnaissance et acquisition automatiques de termes et de leurs variantes en corpus*. Mémoire d'habilitation à diriger des recherches en informatique fondamentale, Université de Nantes

Lame G. (2002), *Construction d'ontologie à partir de textes. Une ontologie du Droit français dédiée à la recherche d'information sur le Web*, thèse de l'Ecole des Mines de Paris

Maedche A. & Staab S. (2000), Mining Ontologies from Text. In *Knowledge Engineering and Knowledge management: methods, models and tools, proceedings of EKAW2000*. R. Dieng and O. Corby (Eds). Bonn : Springer Verlag. LNAI 1937.

Maynard D. et Ananiadou S. (2001), Term extraction using a similarity-based approach, in Bourigault D., Jacquemin C. & L'Homme M.-C., *Recent advances in computational terminology*, John Benjamins Publishing, Amsterdam, pp 261-278

Morin E. (1999), Des patrons lexico-syntaxiques pour aider au dépouillement terminologiques, *Traitemet Automatique des Langues*, volume 40, Numéro 1, pp. 143-166

Nazarenko A., Zweigenbaum P., Habert B. & Bouaud J. (2001), Corpus-based extension of a terminological semantic lexicon, in Bourigault D., Jacquemin C. & L'Homme M.-C., *Recent advances in computational terminology*, John Benjamins Publishing, Amsterdam, pp 327-352

Rastier F. (1991), Sémantique et recherches cognitives, Presses Universitaires de France, Paris, 1991

Rastier F. (1995), Le terme : entre ontologie et linguistique, Actes des 1ères Journées "Terminologie et Intelligence Artificielle", Villetaneuse, avril 1995, *La banque des mots*, Numéro spécial 7-1995, pp. 35-65

Rousselot F., Frath P. et Oueslati R. (1996), Extracting concepts and relations from corpora, *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, workshop on Corpus-Oriented Semantic Analysis, Budapest

Séguéla P. et Aussenac-Gilles N. (1999), Extraction de relations sémantiques entre termes et enrichissement de modèles du domaine, *Actes de la conférence Ingénierie des Connaissances (IC'99)*, Paris, pp 79-88

Slodzian M. (2000), L'émergence d'une terminologie textuelle et le retour du sens, in *Le sens en terminologie*, publication du Centre de Recherche en Terminologie et Traduction de l'Université Lyon 2

Sparck Jones K. (1971), *Automatic Keyword Classification for Information Retrieval*. Butterworth, London

Szulman S., Biébow B. & Aussenac-Gilles N. (2002), Structuration de Terminologies à l'aide d'outils d'analyse de textes avec TERMINAE, *Traitemet Automatique de la Langue (TAL)*. Numéro spécial sur le Structuration de Terminologie. Eds A. Nazarenko, T. Hammon. Vol43, N°1; pp 103-128. 2002.

Toussaint Y., Namer F., Daille B., Jacquemin C., Royauté J. et Hathout N. (1998), Une approche linguistique et statistique pour l'analyse de l'information en corpus. Actes de la 5^{ème} conférence annuelle sur le Traitemet Automatiques des Langues Naturelles (TALN'98), Paris, pp. 182-191

Velardi P., Missikoff M. & Basili R. (2001) Identification of relevant terms to support the construction of domain ontologies. In *ACL WS on Human Language Technologies and Knowledge Management*. Toulouse (F), July 6-7, 2001. 18-28.

Zweigenbaum P. (1999) Encoder l'information médicale : des terminologies aux systèmes de représentation des connaissances. *Innovation Stratégique en Information de Santé* 1999(23)

TALN 2003

Conférences associées

Evaluation des analyseurs syntaxiques
organisée par Laura Monceaux (LIMSI-CNRS)

Atelier sur l'Evaluation des Analyseurs Syntaxiques

Organisé dans le cadre de la conférence TALN'2003

Comité de Programme

Salah Aït-Mokhtar - Xerox, Grenoble
Laura Monceaux - LIMSI, Paris XI
Patrick Paroubek - LIMSI, Paris XI
Jean-Marie Pierrel - LORIA, Nancy
Isabelle Robba - LIMSI, Paris XI
Anne Vilnat - LIMSI, Paris XI

Comité scientifique

Anne Abeillé – LLF, Paris VII
Salah Aït-Mokhtar – Xerox, Grenoble
Philippe Blache – LPL, Aix-en-Provence
Khalid Choukri – ELRA, Paris
John Carroll – University of Sussex, Royaume-Uni
Didier Bourigault – ERSS, Toulouse
Veronique Gendner – TALANA, Paris VII & LIMSI-CNRS, Orsay
Gabriel Illouz – LIMSI, Paris XI
Michèle Jardino – LIMSI, Paris XI
Joseph Mariani – Ministère de la Jeunesse, de l'Education nationale et de la Recherche
Laura Monceaux – LIMSI, Paris XI
Patrick Paroubek – LIMSI, Paris XI
Jean-Marie Pierrel – LORIA, Nancy
Martin Rajman – EPFL, Lausanne
Isabelle Robba – LIMSI, Paris XI
Jacques Vergne – GREYC, Caen
Anne Vilnat – LIMSI, Paris XI
Eric Wehrli – LATL, Genève
Pierre Zweigenbaum – STIM/AP-HP, Paris

Motivation

Depuis une dizaine d'années, avec l'apparition des outils de recherche d'information sur le Web, de nouvelles techniques d'analyse syntaxique plus robustes ont vu le jour. Les analyseurs partiels construisent une analyse parfois minimale, incomplète, mais cela quels que soient la taille et le contenu des données à traiter. D'autre part, les analyseurs qui tentent de produire systématiquement une analyse "complète", ou la plus complète possible continuent d'améliorer leurs résultats.

Devant cette diversité d'offre en matière d'analyseur, il est intéressant voire primordial de proposer une méthodologie permettant de les évaluer. Celle-ci devant inclure :

- La définition d'un format d'annotation permettant une large couverture des phénomènes syntaxiques;
- Le choix d'un corpus et son annotation manuelle (ou semi-automatique) dans ce format d'annotation;
- La définition d'un ensemble de mesures permettant l'évaluation;
- La mise au point des outils aussi bien d'annotation, que de transcription ou d'évaluation.

Pour l'anglais, les métriques et les corpus annotés dans le cadre de la campagne PARSEVAL sont aujourd'hui remis en cause : ils ne sont ouverts ni à d'autres langues ni à de nouveaux formats d'analyse (voir l'atelier de la conférence LREC 2002 : Beyond Parseval towards improved evaluation measures for parsing systems).

Pour le français, la campagne d'évaluation EVALDA/EASY du programme Technolangue (ministère délégué à la recherche et aux nouvelles technologies) qui débute servira de lieu d'expérimentation pour tester de nouvelles approches pour l'évaluation des analyseurs syntaxiques pour le français.

Le but de cet atelier est de développer une réflexion autour des méthodologies d'évaluation, des corpus, des métriques, des outils et des formalismes pour l'évaluation des analyseurs syntaxiques du français.

Programme

Conférence invitée : Anne Abeillé, « Un corpus arboré pour le français : construction et utilisation en évaluation »

Salah Aït-Mokhtar, Caroline Hagège, Agnès Sandor, « Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques »

Sophie Aubin, « Evaluation comparative de deux analyseurs produisant des relations syntaxiques »

Atelier sur l'évaluation des analyseurs syntaxiques

Philippe Blache, « Une grille d'évaluation pour les analyseurs syntaxiques »

Véronique Gendner, Gabriel Illouz, Michèle Jardino, Laura Monceaux, Patrick Paroubek, Isabelle Robba, Anne Vilnat, « Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS »

Gil Francopoulo, « TagChunker : mécanisme de construction et évaluation »

Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques

Salah Aït-Mokhtar, Caroline Hagège, Ágnes Sándor
Xerox Research Centre Europe
6, Chemin de Maupertuis, 38240 Meylan, France
{ait-mokhtar,hagege,sandor}@xrce.xerox.com

Mots-clefs – Keywords

Analyse syntaxique, évaluation, annotation de référence, dépendances, segmentation
Parsing, evaluation, reference annotation, dependencies, segmentation

Résumé - Abstract

Nous discutons dans cet article du protocole d'évaluation des analyseurs syntaxiques, et plus précisément, de la définition de l'annotation de référence en vue d'une évaluation comparative de type «boîte noire», fondée sur les corpus et des structures syntaxiques en dépendances. Nous soulevons les problèmes sous-jacents d'intersubjectivité, à notre connaissance peu abordés dans la littérature. Ces problèmes concernent non seulement la définition des dépendances de référence (Qu'est-ce vraiment un sujet ? un modifieur ? quelle doit être la relation entre typage des dépendances et parties du discours ? etc.) mais aussi la segmentation des textes en mots et phrases. Nous donnons de nombreux exemples qui illustrent ces difficultés et proposons des ébauches de solutions pour certaines d'entre elles.

In this article we discuss basic questions concerning the evaluation of syntactic parsers. We pay special attention to the definition of a reference annotation scheme with syntactic dependency structures for a "black box" comparative evaluation over real-world corpora. We raise the question of intersubjectivity that is little studied in the literature. This problem does not only concern the definition of the dependencies (e.g. What's a subject? What's a modifier? What should be the relationship between dependency types and parts of speech? etc.), but also the segmentation of texts into words and sentences. We give a number of examples that illustrate these difficulties and propose some solutions.

1 Introduction

L'analyse syntaxique consiste à associer des structures grammaticales à des *unités linguistiques*, généralement des *phrases*. Ces structures peuvent être soit de nature *syntagmatique* et décrire des constituants comme les groupes nominaux, verbaux, etc., ou bien de nature *dépendancielle* et décrire les relations grammaticales (ou *fonctionnelles*) entre les mots, comme les relations *sujet, objet*, etc. L'évaluation d'un analyseur consiste alors à mesurer sur des textes réels ses *performances linguistiques*, c'est-à-dire sa capacité à calculer les structures syntaxiques *adéquates* pour des unités linguistiques données¹. Le degrès d'adéquation des structures est établi grâce à la définition préalable d'une annotation syntaxique de référence. On définit également des mesures quantitatives qui rendent compte de façon précise de la distance entre deux structures. Un algorithme d'évaluation calcule alors ces mesures pour un analyseur donné, en comparant les structures que l'analyseur produit aux structures de référence.

Il existe principalement deux méthodologies d'évaluation des analyseurs syntaxiques, qui se distinguent par la nature des données linguistiques utilisées. La première consiste en une évaluation avec un jeu de phrases de test minutieusement sélectionnées (Abeillé, 1991; Lehmann *et al.*, 1996). Les phrases de test sont choisies –souvent créées pour l'occasion, parfois sélectionnées dans des corpus– de telle manière que l'ensemble puisse couvrir le plus grand nombre de constructions syntaxiques. Si une telle évaluation permet de montrer la diversité des phénomènes linguistiques pris en compte par un analyseur, en revanche, elle ne permet pas de prédire son comportement sur des textes réels², comme le montre (Prasad & Sarkar, 2000). D'abord, lorsque les phrases de test sont créées artificiellement, le lexique utilisé est réduit et la distribution des mots n'est pas la même dans les textes. Ensuite, y compris lorsque les phrases sont choisies à partir d'un corpus, elles ne couvrent pas toutes les constructions syntaxiques possibles. En particulier, chaque phrase de test illustre généralement un phénomène spécifique, alors que la combinaison de plusieurs phénomènes complexes est courante dans les textes réels. Enfin, la distribution de ces phénomènes syntaxiques est uniforme dans les jeux de phrases de test, alors qu'elle est très variée dans les textes. Malgré ces limites, les évaluations de ce genre restent très utiles dans le processus de développement et de maintenance des analyseurs.

Nous nous intéressons cependant à la deuxième méthodologie d'évaluation, à savoir celle fondée sur les corpus et qui a émergé avec le projet Parseval (Harrison *et al.*, 1991). Dans le modèle syntagmatique (Harrison *et al.*, 1991), l'analyse de référence est représentée dans un corpus *arboré*, c'est-à-dire des textes où les phrases sont annotées avec des arbres décrivant leurs structures syntagmatiques. Ces corpus de référence sont dans un premier temps créés avec un analyseur particulier (par exemple l'analyseur Fidditch (Hindle, 1994), dans le cas du corpus Penn Treebank (Marcus *et al.*, 1994)) puis corrigés manuellement. Dans le modèle de dépendances (Lin, 1995; Carroll *et al.*, 1998), la référence associe à chaque phrase un ensemble de relations grammaticales, typées ou non, entre ses mots. Le corpus de référence est automatiquement construit à partir de corpus arborés quand ceux-ci existent pour la langue considérée et lorsque les informations nécessaires à une telle transformation y sont explicites (Lin, 1995).

Divers aspects du protocole d'évaluation sont présentés et décrits dans la littérature. On notera par exemple la création de corpus annotés (Marcus *et al.*, 1994; Monceaux, 2002; Abeillé,

¹Les performances computationnelles des analyseurs, déterminantes dans de nombreuses applications, ne seront pas considérées ici.

²Nous utilisons dans cet article les termes *corpus*, *textes réels* ou simplement *textes* pour désigner des textes créés dans le cadre exclusif d'actes de communication.

2003), la transformation d'un corpus arboré en un corpus annoté avec des dépendances (Lin, 1995), la normalisation des sorties hétérogènes de divers analyseurs (Gendner *et al.*, 2002; Monceaux, 2002), la définition des mesures d'évaluation (Harrison *et al.*, 1991; Lin, 1995) ou la définition d'un format d'annotation syntaxique (Ide & Romary, 2003). Nous nous situons dans la perspective d'une évaluation comparative d'analyseurs syntaxiques fondée sur le modèle générale de dépendance, mais indépendante de théories syntaxiques spécifiques. Dans ce qui suit, nous discutons des problèmes d'intersubjectivité, peu étudiés dans la littérature, qui concernent la définition des éléments de base de l'annotation de référence : la nature des dépendances de référence (section 2), ainsi que les notions de mot et de phrase (section 3). Nous montrons par des exemples les aspects qui alimentent les divergences au sujet de ces notions, dont il n'existe pas de définitions rigoureuses et universelles. Il s'agit alors d'élaborer des définitions parfois arbitraires, mais rigoureuses. Leur intérêt sera d'augmenter le degré d'intersubjectivité entre différents annotateurs, de clarifier le processus de normalisation des sorties des analyseurs particuliers par rapport à la référence et d'éviter de les pénaliser sur des choix particuliers différents de ceux, souvent arbitraires, de la référence.

2 La définition des dépendances de référence

Les travaux sur l'évaluation fondée sur les dépendances sont de plus en plus nombreux, mais la littérature aborde peu le problème de la définition rigoureuse et intersubjective des dépendances de référence mises en oeuvre. Une des rares propositions relativement détaillées d'un ensemble de dépendances de référence (relations grammaticales) est celle de (Carroll *et al.*, 2003).

Dans la perspective d'une définition des dépendances rigoureuse et indépendante de théories spécifiques, il reste plusieurs problèmes d'intersubjectivité à résoudre, liés aux aspects que nous présentons dans les paragraphes suivants : le rapport entre dépendances et étiquetage morphosyntaxique, celui entre dépendances et relations sémantiques et, enfin, la modélisation des dépendances de certains éléments textuels spécifiques comme les citations, les listes, etc.

2.1 Dépendances et étiquetage morphosyntaxique

Les définitions des dépendances dans les analyseurs existants ne font pas toujours une distinction stricte entre catégories grammaticales et relations fonctionnelles. L'exemple typique est celui de la relation *modifieur*, souvent définie en fonction de la catégorie syntaxique de la tête. Elle est ainsi éclatée en plusieurs dépendances : *modifieur de verbe*, *modifieur de nom*, *modifieur d'adjectif* et *modifieur d'adverbe*. Cet état des choses fait que les dépendances sont assujetties à l'étiquetage morphosyntaxique effectué par les analyseurs.

Élaborer de telles définitions pour la référence a ses inconvénients. D'abord, cela risque de diminuer le degré d'intersubjectivité entre annotateurs. En effet, s'il est simple de distinguer entre certaines catégories (verbes et noms, par exemple), d'autres sont plus problématiques, à cause de l'absence d'une frontière stricte entre les concepts de catégorie et de fonction syntaxiques. Par exemple, entre *adjectif* et *verbe participe passé* quand celui-ci modifie un nom, les critères de distinction sont moins simples. Ensuite, cela rendrait inutilement plus complexe le processus de normalisation, par rapport à la référence, de la sortie d'un analyseur particulier. Nous proposons par conséquent une évaluation de type «boîte noire» où les dépendances de référence ne devraient pas être définies en fonction de l'étiquetage morphosyntaxique.

2.2 Dépendances et sémantique

Nous discutons dans cette partie de quelques phénomènes à la limite entre dépendances syntaxiques et sémantique, et qui sont déterminants dans les choix des dépendances de référence. Il s'agit de la détection de la tête nominale d'une expression quantifiée, la distinction entre argument et circonstant et, enfin, la définition des sujets (contrôle) des formes verbales non finies.

2.2.1 Tête nominale des expressions quantifiées

Il est nécessaire de connaître la tête d'une expression nominale puisque ces têtes seront les dépendants dans des relations comme celles de sujet, objet et modifieur. Dans le cas des expressions quantifiées, un problème se pose : si on considère que la tête d'une expression comme *trois hommes* est *hommes*, qu'en est-il de l'expression *un million d'hommes* qui, en syntaxe superficielle, correspond à un SN suivi d'un SP ? Et si l'on décide que *hommes* en est également la tête (ce qui semble raisonnable et cohérent compte tenu de l'analyse de *Trois hommes*), alors rien ne devrait s'opposer à en faire de même pour les expressions *beaucoup d'hommes* ou *la plupart des hommes*. Que faire alors dans le cas de *entre trois et cinq de mes hommes, pas mal d'hommes, une multitude d'hommes, un groupe d'hommes*, etc.

Si l'on adopte des critères «de surface», on traite de manière très différente des expressions sémantiquement très proches (par exemple *beaucoup d'hésitations* vs. *maintes hésitations*). Si, en revanche, l'on adopte des critères sémantiques indiquant que la tête d'une expression quantifiée est le nom qui est quantifié indépendamment de la structure de cette expression (un SN seul ou un SN suivi d'un SP), on pourra traiter de manière similaire ces expressions sémantiquement proches. Mais ce faisant, nous devrons alors définir ce qu'est une expression quantifiée de manière intersubjective, ce qui n'est pas aisé. Et probablement, nous nous heurterons à des cas limites comme dans *un troupeau de vaches, une équipe d'ingénieurs*, etc.

2.2.2 Distinction argument/circonstant

Le distinction entre argument (ou *participant*) et circonstant peut être une autre pomme de discorde. Il y a beaucoup de situations intuitivement claires : dans *Jean a mangé une pizza ce midi*, *une pizza* est un argument de *a mangé*, tandis que dans *Jean a mangé avec des baguettes ce midi, avec des baguettes* en est un circonstant. Mais d'autres situations sont plus floues. Par exemple, *pour une banque* est-il argument dans *Jean travaille pour une banque* ? Qu'en est-il de *du nez* dans *Jean saigne du nez* ? Les critères d'ordre syntaxique, comme la possibilité de pronominalisation ou le caractère «obligatoire» de l'argument ne sont pas toujours satisfaisants : dans *Jean décide de partir*, l'expression *de partir* peut être pronominalisée (*Jean le décide*). Tel n'est pas le cas dans *Jean essaie de partir*. Quant au critère du caractère obligatoire de l'argument, il ne s'énonce qu'en fonction des divers «sens» que peut avoir une tête (généralement un verbe). Or, ouvrir la boîte de Pandore qu'est la sémantique lexicale rendrait très complexe la tâche de définition de la référence syntaxique, si tant est que des critères intersubjectifs de sémantique lexicale puissent être élaborés. Nous pensons donc que ces notions d'argument et de circonstant ne sont pas fondées sur des critères universels, hypothèse confirmée par l'observation des entrées verbales dans deux dictionnaires (D_1 et D_2) de patrons syntaxiques, créés manuellement par des lexicographes. Pour les cas «intuitivement flous», le premier dictionnaire semblait avoir des critères plus sélectifs (donc différents). Pour ne citer que quelques exemples : Dans D_2 , un

argument de type *sur+SN* était présent pour *agir* (*agir sur quelque chose*) ; un argument *de+SN* était présent pour *saigner* (*saigner du nez*) ; un argument de type *SN* était présent pour *courir* (*courir 30 mètres en 5 secondes*), etc. Ces constructions étaient toutes absentes de D_1 .

Une fois le rattachement d'un complément (argument ou circonstant) à une tête accompli, la décision de nommer ce rattachement *argument* ou *modifieur* dépend essentiellement du dictionnaire de patrons syntaxiques utilisé. Une évaluation fondée sur cette distinction rendrait impossible une normalisation par rapport à la référence de la sortie des analyseurs évalués qui n'utiliseraient pas le même dictionnaire de patrons syntaxiques.

2.2.3 Contrôles des verbes non finis

Les phénomènes de contrôle du sujet et de l'objet sont bien connus en syntaxe. Les verbes de contrôle sont répertoriés et codés comme tels dans certains lexiques. Si l'on cherche à déterminer, chaque fois que cela est possible, le sujet d'une forme verbale non finie, on va probablement vouloir dépasser l'habituelle acceptation de sujets contrôlés et l'élargir à toute expression présente dans la phrase dont la référence correspondrait au sujet de ce verbe non fini. Ainsi, au même titre que l'on souhaite exprimer que *her* est sujet de *go* dans la phrase *John orders her to go*, on peut également souhaiter exprimer que dans la phrase *John likes her painting flowers* le sujet de peindre *painting* est *her*. Idem pour la phrase *His desire to go was great*, où le sujet de *go* correspond à la référence du possessif *His*. Dans ce dernier cas, la question est de savoir si l'on exprime l'existence d'une relation sujet entre *go* et *His*.

2.3 Quelles dépendances pour les éléments textuels spécifiques ?

Les textes réels contiennent nombre d'éléments ayant des caractéristiques linguistiques et extra-linguistiques (typographiques, graphiques ou de rendement visuel) qui sont très peu étudiées et prises en compte en traitement automatique. Ces éléments distingués peuvent pourtant y être présents fréquemment, comme le montre (Gala Pavia, 2003). Parmi ces éléments, on peut citer les titres et les sous-titres, les listes, les tables, les expressions mises entre parenthèses ou guillemets (et parmi ces dernières les citations). Quelles dépendances mettent en jeu ces éléments et comment les définir dans le cadre d'une annotation de référence ?

Prenons l'exemple des citations directes, souvent mises entre guillemets. On peut penser que de tels éléments, représentés par leur tête syntaxique, se rattachent simplement comme compléments (de type argumental) des verbes qui les introduisent. Dans l'exemple suivant :

(1) «*Je meurs de faim*», dit-il.

on rattacherait la tête verbale *meurs* à *dit* dans une dépendance de type complément. Mais dans bien des cas, la situation est plus complexe. D'abord, plusieurs rattachements d'éléments internes à la citation sont possibles. Dans l'exemple qui suit, la citation est à la fois le complément de *dit*, mais aussi une conjonction de subordination pour la phrase précédente :

(2) Beaucoup se demandent si le rôle des enseignants est de commenter l'actualité en classe...«*Mais, quand la demande est là, on ne peut pas faire comme si elle n'existe pas*», dit un professeur d'histoire à Etampes (Essonne). (Libération)

Évidemment, les rattachements peuvent se faire dans le sens inverse. Dans l'exemple suivant, la *Focus RS* est le sujet de *permet* (et donc aussi le «sujet» de *donner*), alors que *mettre* est coordonné avec *donner* et *Selon lui* est modifieur de *permet* :

- (3) Selon lui, la Focus RS, qui ne sera vendue qu'à 300 exemplaires par an en France, "permet aussi de donner du rêve" et de mettre en valeur la gamme Ford. (Le Monde)

Il est intéressant de noter que la citation peut avoir comme point d'ancrage syntaxique un verbe généralement admis comme intransitif. Faut-il dans l'exemple suivant considérer la citation comme un complément argumental direct du verbe *s'épancher* ?

- (4) A la barre, il s'épanche un peu plus : «*La plupart des agences font appel à ce système et à cette société londonienne. C'est une pratique courante.*» (Libération)

La proposition principale qui introduit la citation, ainsi que ses éventuels modificateurs, peut s'insérer à l'intérieur de l'espace délimité par les guillemets, une police de caractères particulière permet alors de la distinguer de la citation. Cette observation pose la question du format des textes du corpus de référence, et de la décision, généralement prise, d'en éliminer les diverses balises de marquage à valeur structurelle ou de rendement visuel :

- (5) "*C'est vrai, reconnaît un responsable français, c'était plein d'ambiguïtés. Mais on se disait, on verra, demain est un autre jour.*" (Le Monde)

L'utilisation du marquage typographique facilite aussi l'insertion d'expressions linguistiques dans une autre langue que celle, principale, du texte. Quelles dépendances (et quelle segmentation en mots) représenter alors ?

- (6) «*Nemo auditur turpitudinem allegans*» (Nul ne peut se prévaloir de ses propres turpitudes), rétorque, en latin, le procureur. (Le Monde)

Enfin, signalons l'existence, toutefois plus rare, de citations imbriquées :

- (7) ""Mal nommer les choses, c'est ajouter du malheur au monde", *disait Camus*", rappelle cet érudit passionné. (Le Monde)

Ces difficultés concernant les citations valent aussi pour d'autres éléments textuels particuliers comme les listes, dont le cas est étudié dans (Gala Pavia, 2003; Aït-Mokhtar *et al.*, 2003). Si l'on désire faire une évaluation sur des corpus variés et non modifiés pour l'occasion, il est nécessaire que la référence modélise ces structures spécifiques.

2.4 Un exemple de critères pour une annotation de référence

Dans cette section, nous illustrons la manière dont nous avons procédé pour définir un guide d'annotation de référence pour des analyseurs syntaxiques du français et de l'anglais. Le but poursuivi est de fournir des critères objectifs aux annotateurs afin qu'ils puissent procéder à l'annotation de textes avec un fort degré (idéalement total) d'intersubjectivité. Afin d'orienter le traitement syntaxique que nous effectuons, nous avons ciblé les applications d'extraction d'informations. Ainsi, à partir des dépendances que nous extrayons dans les textes, nous devrons être capables de répondre plus facilement aux questions "qui fait quoi, où, comment, quand, pourquoi et avec qui/quoi". Nous avons fait des choix qui peuvent être considérés comme discutables du point de vue de la tradition syntaxique. D'une manière générale, nous avons préféré prendre une position, même sujette à controverse, que de laisser des zones de flou qui rendent difficile la tâche d'annotation et d'évaluation.

Dans la mesure où il serait trop long dans cet article de donner tous les détails des critères de détermination de toutes les dépendances considérées, nous nous contenterons de détailler notre proposition du repérage de la dépendance SUJET. Nous expliciterons dans un premier temps quels sont les critères qui nous permettent de détecter le sujet d'un verbe fini et dans un deuxième temps nous verrons ce que nous considérons comme sujet de verbes à des formes non finies (contrôle), avec des exemples pour le français et pour l'anglais.

2.4.1 Sujets de verbes finis

Les sujets des verbes finis sont repérés de la manière suivante : Si l'on a une proposition contenant un verbe fini, le sujet de ce verbe fini est l'expression linguistique dont le référent instancie *Qui* ou *Qu'est-ce que qui* lorsque cette proposition est mise à la forme interrogative. L'expression en question peut être pleine ou se limiter à une reprise anaphorique (pronom), à l'exclusion des pronoms clitiques non sujet dépendants du verbe considéré. Pour la phrase *Faire du sport est une bonne chose pour la santé*, l'interrogative correspondante est *Qu'est-ce qui est une bonne chose pour la santé ?* et l'expression qui instancie *qu'est-ce qui* est *Faire du sport*. Le sujet de *est* est donc *Faire du sport*. Selon ce critère, le sujet d'un verbe à la forme passive sera toujours le sujet de surface. Dans le cas des relatives sujet, le sujet sera le pronom relatif.

2.4.2 Sujets de verbes non finis

Le critère de détection du sujet de verbes non finis est similaire au critère général de la détection du sujet des verbes finis. Pour la phrase *Je vois Marie venir*, le sujet de *venir* est *Marie* dans la mesure où c'est l'expression *Marie* qui instancie *Qui* dans *Qui vient ?*. De même, dans *Jean parle en dormant*, on considérera que *Jean* est non seulement sujet du verbe fini *parle* (*Qui parle ?*) mais aussi du participe présent *dormant* (*Qui dort ?*).

Cependant, on module ce critère dans le cas d'infinitives qui sont introduites par des verbes exprimant ordre, volition, modalité ou accompagnés de négation. Dans *Pierre ordonne à Jean de venir*, on ne peut pas répondre à la question *Qui vient ?* (Jean n'est pas encore venu et nous ne savons pas s'il viendra). En revanche, si nous reformulons la question de la manière suivante : *Qui vient si ce que Pierre ordonne se réalise*, la réponse est bien *Jean*. Le critère de détection des sujets d'infinitives est donc élargi en admettant que nous pouvons compléter l'interrogative en supposant l'accomplissement d'une circonstance introduite par le verbe principal.

Comme pour les verbes finis, nous admettons que le sujet d'un verbe non fini puisse être une expression référentielle. Ainsi, dans *La seul reproche que Marie lui fait, c'est d'être parti sans le lui dire*, le sujet de *être parti* et de *dire* est le premier *lui* de la phrase car c'est la seule expression du texte qui puisse instancier les interrogatifs de *qui est parti ?* et *qui le lui a dit ?*. De même, nous admettons qu'un possessif puisse être sujet. En effet, dans une phrase comme *Sa volonté de vivre l'a sauvé*, nous considérons comme sujet de *vivre* le possessif *Sa*. À la question *Qui vit ?* nous ne pouvons certes pas trouver dans la phrase une expression définie qui réponde à cette question, mais nous savons que *sa* réfère à l'entité dont le nom répondrait à la question. Pour des raisons similaires, on devra annoter que *her* est le sujet de *painting* dans *John likes her painting flowers* (où *flowers* est l'objet de *painting*.)

3 La segmentation en mots et en phrases

Les méthodes d'évaluation supposent une référence formée d'une suite de *phrases* syntaxiquement annotées, chaque phrase étant une suite d'unités de base (généralement des *mots*, mais aussi des symboles de ponctuation). Dans une évaluation de type dépendanciel, la référence décrit les relations grammaticale qui doivent être calculées entre les mots, à l'intérieur d'une même phrase. L'annotateur doit donc décider de la segmentation du texte en mots et en phrases.

3.1 Segmentation en mots

La segmentation en mot n'est pas chose triviale, notamment à cause des rôles ambigus des « séparateurs » (le blanc, l'apostrophe et le tiret). Leur présence est parfois arbitraire (*parce que* et *lorsque*) et, malheureusement, n'implique pas une frontière entre mots (*aujourd'hui*, *rendez-vous*, *pomme de terre*). (Grevisse, 1993) va même jusqu'à proposer de considérer les formes d'un temps composé (*a vendu*) comme une forme unique, malgré les possibilités d'insertion (*il a, selon le témoignage de ses proches, vendu la voiture à un voisin.*) Notons également que, au moins dans une perspective de traitement automatique, la présence de séparateurs n'est pas nécessaire pour délimiter les mots (au sens d'unités de base pour l'analyse syntaxique) : dans les constructions productives (non lexicalisées) avec des préfixes (*retester*), il est utile de considérer séparément les deux unités (*re* et *tester*), afin de construire dynamiquement le sens de l'ensemble. On le voit, la segmentation en mots dépend du dictionnaire morphologique utilisé et des stratégies choisies pour le traitement des mots composés.

Dans cette situation, quels critères doit-on définir pour garantir une segmentation en mots cohérente dans la référence, faciliter la tâche des annotateurs et tolérer des segmentations différentes pour les systèmes à évaluer ? Il semble raisonnable d'opter, comme dans le projet GRACE (Adda *et al.*, 1999), pour une segmentation en mots minimaliste. Un critère très clair existe pour ce choix : il s'agit de découper les mots partout où des symboles de séparateurs (au moins le blanc, l'apostrophe et le tiret) apparaissent. Certes, des segments seront inutilement découpés en plusieurs unités (*aujourd'hui* en *aujourd'*, *'* et *hui*) mais la référence peut les lier avec des dépendances spécifiques. L'avantage de ce critère est que la segmentation de référence peut alors être entièrement automatisée et une cohérence totale est garantie. Lorsque, pour un analyseur donné, une différence de segmentation avec la référence est détectée, l'algorithme d'évaluation ne doit pas considérer les dépendances internes au segment litigieux.

3.2 Segmentation en phrases

Le problème de l'intersubjectivité concernant la segmentation en phrases est autrement plus complexe. On peut « définir » la phrase comme l'unité minimale de communication linguistique (Grevisse, 1993), mais cela aurait comme conséquence d'exclure les phrases comportant des références anaphoriques : elles ne sont pas des unités minimales de communication puisque leur interprétation dépend des unités textuelles précédentes. Si on considère la phrase comme une unité linguistique « suivie d'une pause importante », le problème est alors de déterminer ces pauses dans les textes. Elles sont généralement représentées par un point, mais aussi par les points de suspension, le point d'interrogation, le point d'exclamation, le point-virgule, le double point. Selon (Grevisse, 1993), « *La virgule peut même séparer des phrases, que nous appelons sous-phrases dans ce cas.* » L'absence du point n'implique donc pas la continuité de ce qui est intuitivement appelé « phrase ». À l'inverse, ces signes de ponctuation ne marquent pas toujours une discontinuité syntaxique, comme le montrent les exemples suivants :

(8) Je vais à la pêche avec toi ! cria-t-il. (Colette)

(9) Le prochain rendez-vous de la section est prévu le 5 mai. Pour voter. (Le Monde)

Par ailleurs, on trouve dans les corpus nombre d'éléments textuels qui ne sont pas toujours ponctués (par exemple les titres et sous-titres) ou qui contiennent des ponctuations diverses

(point, point-virgule, double-point) mais qui constituent pourtant des unités syntaxiques entières. Parmis ces dernières, on peut citer les listes, assez fréquentes dans les corpus scientifiques, juridiques ou techniques, ou bien les citations, extrêmement fréquentes dans les textes journalistiques. Les exemples suivants illustrent notre propos :

(10) Déposer :

- les roues.
- le manchon d'entrée d'air.
- le vase d'expansion, le fixer sur le moteur.

(Manuel de maintenance)

(11) *"Pour moi, leur élimination n'avait rien à voir avec le foot mais avec ça, assure Avraham Grant en se tapotant la tête. Ils étaient si arrogants".* (Le Monde)

Notons que les signes de ponctuation utilisés dans ces éléments textuels ne sont pas toujours les mêmes et que parfois, ils y sont absents (par exemple, nous avons trouvé beaucoup de listes dont les items ne se terminaient par aucun signe de ponctuation.) Des critères minimalistes de segmentation en phrases, purement typographiques, auraient le double inconvénient d'ignorer nombre de relations syntaxiques principales (sujet, objet, modifieur) qui s'étalent au-delà des signes de ponctuation, et aussi de donner lieu à des incohérences de segmentation à causes de l'irrégularité observée de la ponctuation. Les exemples de citations comme (4), (5) et (11) montrent que si l'on veut lier toute l'expression linguistique de la citation à l'élément qui l'introduit (généralement un verbe de communication), il faut lier les différentes phrases (ou sous-phrases) qui la composent, même lorsque celles-ci sont séparées par des ponctuations «fortes». On pourrait représenter ce lien par une dépendance de type *coordination* ou *séquence*. Nous proposons donc que la référence établisse des dépendances même entre des mots séparés par des ponctuations fortes, pourvu qu'il y ait une continuité syntaxique. Comment définir celle-ci ? Évidemment, on exclue les liens anaphoriques ou ontologiques (hypéronymie, méronymie, etc.) La continuité syntaxique se vérifie uniquement grâce aux définitions des dépendances de référence, élaborées indépendamment de la ponctuation. Par exemple, chacun des items de la liste (10) devrait être lié dans une dépendance de type *objet* au verbe *Déposer*. De même, dans l'exemple (9), *voter* serait modifieur de *est prévu*.

4 Conclusion

Dans cet article, nous avons discuté des problèmes d'intersubjectivité dans la définition d'une annotation de référence pour l'évaluation des analyseurs syntaxiques fondée sur les dépendances. Ces problèmes surgissent surtout lors de l'évaluation comparative de systèmes utilisant des ressources et des modélisations linguistiques différentes. Il est nécessaire de définir avec précision l'ensemble des dépendances de référence, dans le but d'avoir une annotation cohérente, de simplifier et clarifier le processus de normalisation des sorties des systèmes à évaluer, et de ne pas les pénaliser sur des choix de ressources ou de descriptions linguistiques particulières. Nous avons également souligné les divergences qui concernent la segmentation en mots et en phrases. Nous proposons de ne pas prendre en compte la ponctuation pour définir l'espace des expressions linguistiques sur lequel les dépendances doivent être calculées, mais l'inverse : la définition précise des dépendances de référence devrait se faire indépendamment de la ponctuation traditionnellement considérée comme indiquant les fins de phrases. L'évaluation du rappel des analyseurs en sera plus exigeante, mais elle aura l'avantage de mieux rendre compte de leur capacité à extraire les relations syntaxiques dans les textes réels.

Références

- ABEILLÉ A. (1991). Analyseurs syntaxiques du français. *Bulletin semestriel de l'ATALA (Association pour le traitement automatique des langues)*, 32(2).
- A. ABEILLÉ, Ed. (2003). *Building and Using Parsed Corpora*. Dordrecht: Kluwer.
- ADDA G., MARIANI J., PAROUBEK P., RAJMAN M. & LECOMTE J. (1999). L'action GRACE d'évaluation de l'assignation de parties du discours pour le français. *Langues*, 2(2), 119–129.
- AÏT-MOKHTAR S., LUX V. & BÁNIK E. (2003). Linguitic parsing of lists in structured documents. In *Proceedings of the 3rd Workshop on NLP and XML (NLPXML-2003)*, Budapest.
- CARROLL J., BRISCOE T. & SANFILIPPO A. (1998). Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, p. 447–454, Granada, Spain.
- CARROLL J., MINNEN G. & BRISCOE T. (2003). Parser evaluation: Using a grammatical relation annotation scheme. In A. ABEILLÉ, Ed., *Building and Using Parsed Corpora*, p. 299–316. Dordrecht: Kluwer.
- GALA PAVIA N. (2003). *Un modèle d'analyseur syntaxique robuste basé sur la modularité et la lexicalisation de ses grammaires*. PhD thesis, Université Paris XI.
- GENDNER V., ILLOUZ G., JARDINO M., MONCEAUX L., PAROUBEK P., ROBBA I. & VILNAT A. (2002). A protocol for evaluating analyzers of syntax (PEAS). In *Proceedings of the Language and Resources Evaluation Conference (LREC)*, Las Palmas.
- GREVISSE M. (1993). *Le bon usage*. Paris - Louvain-la-Neuve: Duculot.
- HARRISON P., ABNEY S., BLACK E., FLICKINGER D., GDANIEC C., GRISHMAN R., HINDLE D., INGRIA B., MARCUS M., SANTORINI B. & STRZALKOWSKI T. (1991). Evaluating syntax performance of parser/grammars of English. In *Proceedings of the Workshop on Evaluating Natural Language Processing Systems*, p. 71–77, Berkeley, CA.
- HINDLE D. (1994). A parser for text corpora. In B. ATKINS & A. ZAMPOLLI, Eds., *Computational Approaches to the Lexicon*. New York: Oxford University Press.
- IDE N. & ROMARY L. (2003). Encoding syntactic annotation. In A. ABEILLÉ, Ed., *Building and Using Parsed Corpora*, p. 281–296. Dordrecht: Kluwer.
- LEHMANN S., OEPEN S., REGNIER-PROST S., NETTER K., LUX V., KLEIN J., FALKEDAL K., FOUVRY F., ESTIVAL D., DAUPHIN E., COMPAGNION H., BAUR J., BLAKAN L. & ARNOLD D. (1996). TSNLP test suites for natural language processing. In *Proceedings of the International Conference on Computational Linguistics (COLING-96)*, p. 711–716, Copenhagen.
- LIN D. (1995). A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, p. 1420–1425, Montréal.
- MARCUS M. P., SANTORINI B. & MARCINKIEWICZ M. A. (1994). Building a large annotated corpus of English: the Penn treebank. In S. ARMSTRONG, Ed., *Using Large Corpora*. MIT Press.
- MONCEAUX L. (2002). *Adaptation du niveau d'analyse des interventions dans un dialogue : Application à un système de question-réponse*. PhD thesis, Université Paris XI.
- PRASAD R. & SARKAR A. (2000). Comparing test-suite based evaluation and corpus-based evaluation of a wide-coverage grammar for english. In *Proceedings of the LREC'2000 Workshop on Using Evaluation within HLT Programs: Results and Trends*, p. 7–12, Athena, Greece.

Evaluation comparative de deux analyseurs produisant des relations syntaxiques

Sophie Aubin

Unité Mathématiques, Informatique et Génome
Institut National de la Recherche Agronomique
78352 Jouy-en-Josas
sophie.aubin@jouy.inra.fr

Mots-clefs – Keywords

Evaluation, Link Parser , analyseur hybride, relations syntaxiques, système d'EI
Evaluation, Link Parser, syntactic dependencies, Information Extraction system

Résumé - Abstract

Nous présentons dans cet article un exemple d'évaluation comparative entre deux analyseurs syntaxiques pour l'anglais. La particularité de ce travail est que ce sont les relations syntaxiques qui sont considérées et non le découpage syntagmatique des énoncés. Nous proposons une méthode qui utilise la distinction de divers phénomènes linguistiques, affinant ainsi la précision de l'évaluation. Nous présentons et commentons les résultats de cette évaluation.

We describe in this paper the comparative evaluation of two syntactic parsers for English. We considered syntactic relations instead of constituency. We propose a method that allows the distinction between different linguistic phenomena and we present the results of this evaluation.

1 Introduction

Les méthodes utilisées pour l'évaluation des outils d'analyse syntaxique sont nombreuses mais les protocoles à mettre en place pour y parvenir ne sont pas toujours clairement définis ni adéquats. Ceci est particulièrement vrai pour le français, comme le montre (Gendner *et al.*, 2002). Il semblerait que les tests soient souvent dictés par le contexte de travail et le but recherché par celui qui procède à cette évaluation, ce qui est notre cas. Le travail que nous présentons est placé dans un cadre tout à fait particulier et ne peut donc pas aboutir à la construction d'un modèle. Nous souhaitons plutôt, à travers cet article, expliquer les choix que nous avons faits durant les tests et espérons avancer des pistes pour des évaluations futures. Le point de vue principal que nous défendrons est celui qui tend à montrer l'intérêt de faire porter l'évaluation sur les relations syntaxiques et non sur le découpage syntagmatique comme c'est généralement le cas, par exemple lors de la campagne d'évaluation PARSEVAL, (Black *et al.*, 1991).

Plusieurs caractéristiques de notre évaluation découlent du contexte dans lequel elle a été menée.

La première particularité est qu'il s'agit d'un test comparatif entre deux analyseurs présentant des stratégies différentes. Le premier est un analyseur en dépendances : Link Parser, le second est un logiciel commercial basé sur l'analyse en constituents et en dépendances (dit "hybride"). Nous l'appellerons ACD.

La seconde caractéristique concerne les données de test qui sont assez particulières puisqu'elles combinent deux méthodes mises en opposition dans (Srinivas *et al.*, 1998) et (Prasad & Sarkar, 2000) : l'utilisation d'un corpus "naturel" ("corpus-based" en anglais) et celle d'un corpus de test (ou "test-suite based"). Ainsi, notre ensemble de test, constitué de phrases extraites de notre corpus d'origine (des résumés d'articles scientifiques traitant de génomique fonctionnelle provenant de la base Medline), est organisé de façon à identifier des phénomènes linguistiques particuliers. Ceci nous permet d'évaluer la construction des relations syntaxiques (sujet-verbe, nom-complément prépositionnel, etc.) plutôt que le découpage syntagmatique. Ce choix est dicté par notre système d'extraction d'information dans lequel doit s'insérer l'analyseur syntaxique. Le logiciel Asium¹, qui utilise les résultats de l'analyse, construit les cadres de sous-catégorisation des mots ainsi que des classes sémantiques à partir des arguments et ajouts des noms et verbes du corpus, voir (Faure & Nédellec, 1999).

Enfin, notre corpus étant principalement composé de phrases longues (27 mots par phrase en moyenne), nous avons choisi d'évaluer la qualité des relations extraites à partir de phrases présentant des phénomènes linguistiques complexes, axant ainsi notre évaluation sur les qualités de robustesse des analyseurs.

2 Présentation des analyseurs

Les systèmes mis en comparaison sont tous deux des analyseurs syntaxiques robustes : ils fournissent une analyse pour toute phrase passée en entrée, sans que l'on note une baisse brutale de sa qualité avec l'augmentation de la complexité de l'énoncé. Les grammaires utilisées sont toutefois très différentes, mettant en oeuvre deux mouvements de l'analyse syntaxique : l'analyse en constituants et/ou l'analyse en dépendances, voir (Schneider, 1998).

¹http://www.lri.fr/~faure/Demonstration/Presentation_Demo.html

2.1 Définition des exigences

Nous utilisions ACD au sein d'un système plus vaste d'extraction d'information et cherchions à le remplacer. Avant même les tests, nous avons donc dû considérer plusieurs critères : le nouvel analyseur devait traiter au moins l'anglais, être robuste et produire des résultats compatibles avec notre logiciel de classification ASIUM qui exploite des relations typées binaires du type $REL(w_1, w_2)$.

En plus de ces exigences, nous avions des souhaits particuliers comme la disponibilité immédiate, la gratuité, ainsi que la possibilité de faire évoluer la grammaire et les dictionnaires de l'analyseur. Nous n'avions cependant pas de contraintes sur le temps de traitement.

LP répondant à toutes ces exigences et souhaits, nous l'avons sélectionné (contrairement à d'autres comme FDG de Connexor² ou l'analyseur du Greyc³) pour les tests. Un autre analyseur avait été retenu ; il s'agit de Minipar, développé par Dekang Lin, Université d'Alberta, Canada⁴, voir (Lin, 1998). Ses performances étant équivalentes à celles de LP, mais sa documentation beaucoup moins riche et son formalisme moins accessible, nous ne l'avons pas gardé.

2.2 L'analyseur ACD

ACD est ce qu'on appelle un système hybride : à partir de l'analyse syntagmatique (*i.e.* en constituants), il procède dans un second temps à la construction de relations syntaxiques entre les mots (ou dépendances) du type “*SUBJ(residue contain)*”. Ces deux étapes sont liées, la seconde reposant sur la première (deux mots ne peuvent être en relation que s'ils font partie d'un même groupe de même niveau syntagmatique).

La grammaire définit les unités syntagmatiques par des contraintes (pas seulement par des suites d'étiquettes morpho-syntaxiques) comme la nécessité de l'accord en nombre entre le sujet et le verbe. Ces règles sont ordonnées suivant leur degré de généralité dans la langue. Ainsi, l'obligation de l'accord sujet-verbe est donnée par les règles de plus haut rang. Des règles de rang inférieur prévoient des cas où cette contrainte n'est pas remplie dans la langue. Par exemple, “*On [sing.] est rentrés [plur.]*”. Si aucune contrainte n'a été satisfaite, le segment qui n'a pas pu être analysé est laissé en l'état. Ceci assure la robustesse du système.

2.3 Link Parser

Link Parser⁵ (LP) a été développé au sein de l'Université Carnegie Mellon (Pittsburgh, USA) par Daniel Sleator, Davy Temperley et John Lafferty. LP repose sur une grammaire de dépendances, Link Grammar, inspirée des travaux de Lucien Tesnières : pour chaque phrase, le système construit une analyse syntaxique à partir de tous les liens créés localement entre paires de mots. Chaque lien entre deux mots (dépendance) est typé (sujet-verbe, adjectif-nom, etc.) et répond à des contraintes décrites dans le dictionnaire. Les mots y sont en effet organisés par type de comportement (par exemple verbes transitifs ou non). A chaque famille de mots s'appliquent des contraintes particulières définies dans des règles. La documentation fournie avec l'analyseur, (Temperley, 1999), offre une description et une explication préciseuses de la

²http://www.connexor.com/m_syntax.html

³<http://users.info.unicaen.fr/~giguet/java/textes/index.html>

⁴<http://www.cs.ualberta.ca/~lindek/downloads.htm>

⁵<http://www.link.cs.cmu.edu/link/>

grammaire utilisée. L'algorithme, que nous ne présenterons pas ici, est expliqué dans (Sleator & Temperley, 1991).

Le résultat de l'analyse est une suite de relations binaires, qui, assemblées, permettent une visualisation de l'analyse sous la forme d'un graphe d'arcs planaires. Une représentation syntagmatique de la phrase est aussi produite. Dans l'exemple suivant, les colonnes de gauche et de droite contiennent les mots mis en relations par le lien dont l'étiquette apparaît au centre ("Spx" : sujet, "Pvf" : auxiliaire-passif, "TOf" : verbe-to, "I" : to-infinitif, "Os" : verbe-objet.).

<u>residues.n</u>	Sp	←	Spx	→	Spx	are.v
are.v	Pv	←	Pvf	→	Pvf	known.v
known.v	TOf	←	TOf	→	TO	to
to	I	←	I	→	I	<u>contain.v</u>
contain.v	O	←	Os	→	Os	activity.n

Cet ensemble de mots et de liens doit être traité *a posteriori* pour fournir par exemple la relation **SUJET(contain, residue)** qui nous intéresse. Nous avons développé un programme (*SortieLP.java*) qui se charge de cette tâche.

LP présente des caractéristiques intéressantes : il est gratuit et directement accessible par téléchargement via Internet. Sa documentation est riche et permet une bonne utilisation du système. Le source (en C), et le dictionnaire-grammaire sont fournis et modifiables. Enfin, l'insertion d'une nouvelle règle dans le dictionnaire-grammaire ne compromet pas l'ensemble de l'analyse. Link Parser a été développé pour l'anglais.

3 Méthodes et métrique

3.1 Corpus de test

Il est assez difficile d'évaluer la qualité de l'analyse d'une phrase complète, particulièrement lorsqu'elle n'est pas écrite dans notre langue maternelle et qu'elle traite, de plus, d'un domaine qui ne nous est pas familier, comme la génomique. De nombreuses ambiguïtés (réelles ou apparentes) se glissent dans ce type de textes techniques dans lesquels les phrases sont souvent longues et présentent parfois, entre autres difficultés, plusieurs coordinations et relatives.

Ne disposant que d'assez peu de temps pour choisir un analyseur parmi d'autres, nous avons pris le parti de n'évaluer que des sous-parties d'analyses, correspondant à des relations syntaxiques particulières. L'intérêt de cette méthode est qu'elle nous permettait d'avoir une vision précise des performances de chaque analyseur dans différentes situations (détection de relations verbales ou nominales, phrases simples et phrases longues ou complexes, etc.). Nous pensions alors particulièrement aux coordinations et aux phrases présentant plusieurs propositions.

Nous avons donc constitué un corpus divisé en 22 fichiers de 10 phrases correspondant à autant de phénomènes linguistiques que nous souhaitions tester : détection de sujet, COD, attachement des prépositions, coordinations, etc. Ces choix ont été motivés en partie par la nature du corpus récupéré via Internet sur Medline : phrases longues, nombreuses coordinations et formes passives, nombreux mots inconnus des analyseurs.

Comme il est démontré dans (Prasad & Sarkar, 2000), il est préférable d'évaluer un système à l'aide de deux types de données : un ensemble extrait d'un corpus existant et un ensemble de phrases de test créées artificiellement. Le premier groupe de données doit permettre d'évaluer le nombre de phrases et le second le nombre de constructions syntaxiques que l'analyseur est capable de couvrir. Pour des raisons de temps, nous avons opté pour un compromis entre les deux

en ne constituant qu'un seul jeu de données : nous avons choisi des phrases dans notre corpus de base et nous les avons classées selon leur contenu et leur intérêt syntaxiques. Les énoncés ne sont donc pas artificiels tout en rendant compte de phénomènes linguistiques particuliers. La liste de ces phénomènes est présentée dans le tableau 1.

3.2 Métrique

Pour chaque type de construction (Sujet, Nom *of* Nom, etc.) nous avons relevé manuellement les relations qui nous intéressaient pour chaque phrase. Comme on trouve souvent plusieurs relations du même type dans une phrase, les notes maximales sont au moins égales à 10 (nombre de phrases par fichier) et varient d'un groupe à l'autre, comme on peut le voir dans la colonne *nbRel* du tableau 1.

A partir des notes de l'évaluation (colonnes *relOK* et *relTot*), nous avons calculé le *rappel* et la *précision* pour groupe linguistique traité par chacun des deux analyseurs.

$$rappel = \frac{\text{relations pertinentes produites}}{\text{relations pertinentes totales}} \quad \text{ou dans le tableau 1} \quad rappel = \frac{\text{relOK}}{\text{nbRel}}$$

$$\text{précision} = \frac{\text{relations pertinentes produites}}{\text{relations produites totales}} \quad \text{ou dans le tableau 1} \quad \text{précision} = \frac{\text{relOK}}{\text{relTot}}$$

4 Résultats et commentaires

Les résultats pour les deux analyseurs sont réunis dans le tableau 1. La seconde colonne indique le **type de la construction** considérée dans le test. On indique ensuite les notes avec le **rappel** et la **précision** définis dans la section précédente.

Nous ne commenterons que les principales constructions : sujet-verbe, verbe-COD, les groupes prépositionnels, les Nom *OF* Nom en particulier, la coordination, la négation, les adjectifs comparatifs, le passif, les propositions relatives. Certaines constructions regroupent plusieurs tests concernant des phénomènes proches. La première colonne du tableau est un renvoi au paragraphe de commentaire.

1. le sujet : les deux systèmes présentent des résultats similaires et satisfaisants (rappel autour de 0.75), si on considère le fait qu'ils n'ont pas été adaptés au domaine. Les quelques erreurs rencontrées sont dues à une mauvaise analyse complète de la phrase ou à une mauvaise détection du verbe. D'une manière générale, la relation sujet-verbe, étant parmi les plus simples, ne peut pas être considérée comme très significative de l'efficacité d'un analyseur syntaxique.

2. le COD : bien que le sujet et l'objet aient tous deux une position similaire autour du verbe, les résultats sont très différents. ACD échoue de façon assez surprenante dans le traitement de phrases pourtant simples où certains COD juxtaposés au verbe ne sont pas détectés. Par exemple : “*The trp RNA-binding attenuation protein (TRAP) regulates[V] expression[OBJ] of the B. subtilis trp operon ...*”. LP présente pour sa part de meilleurs résultats. Notons que LP considère le sujet et l'objet à un même niveau par rapport au verbe, contrairement à ACD.

§	Relation	nbRel	Link Parser				ACD			
			relOK	rapp.	relTot	préci.	relOK	rapp.	relTot	préci.
1	sujet	18	13	0.72	19	0.68	14	0.78	20	0.65
2	COD	18	16	0.89	17	0.94	9	0.5	13	0.69
3	Prep	48	25	0.52	55	0.45	20	0.42	49	0.41
	V-GP1	14	13	0.93	15	0.87	9	0.64	23	0.39
	V-GP2	14	11	0.76	15	0.73	9	0.64	26	0.35
	O-GP	16	7	0.43	12	0.58	12	0.75	28	0.43
	NofN	16	13	0.81	15	0.87	14	0.87	26	0.54
	VtoV	10	9	0.9	9	1	7	0.7	7	1
4	VcooV	10	8	0.8	9	0.89	6	0.6	6	1
	NcooN	10	8	0.7	10	0.8	4	0.4	6	0.67
	mucoo	29	11	0.38	20	0.55	10	0.34	18	0.56
	boAnd	11	7	0.64	10	0.7	3	0.27	7	0.43
	eitOr	11	7	0.64	10	0.7	3	0.27	12	0.25
	aucoo	11	7	0.64	10	0.7	6	0.54	11	0.54
5	nV-Adj	10	8	0.8	9	0.89	0	0	0	1
	nN-GP	10	4	0.4	4	1	0	0	0	1
6	moThan	10	5	0.5	9	0.55	0	0	2	0
	erThan	11	8	0.73	9	0.89	3	0.27	7	0.43
7	PaSim	18	17	0.94	18	0.94	17	0.94	22	0.77
7, 8	PaRel	12	11	0.92	11	1	8	0.67	11	0.73
8	SuRel	19	17	0.89	18	0.94	14	0.74	18	0.78
	CoRel	10	7	0.7	7	1	1	0.1	2	0.5

Détail des abréviaitons de noms de relations :

sujet = sujet-verbe, **COD** = verbe-objet direct, **Prep** = groupe prépositionnel quelconque, **V-GP** = verbe-groupe prépositionnel, **O-GP** = Objet-groupe prépositionnel, **NofN** = Nom of Nom, **VtoV** = Verbe to Verbe, **VcooV** = Verbe coordination Verbe, **NcooN** = Nom coordination Nom, **mucoo** = coordinations multiples, **boAnd** = both ... and, **eitOr** = either ... or, **aucoo** = autres coordinations, **nV-Adj** = not Verbe ou Adjectif, **nN-GP** = not Nom ou groupe prépositionnel, **moThan** = more ... than, **erThan** = -er ... than, **PaSim** = passif simple, **PaRel** = passif-relative, **SuRel** = sujet-relative, **CoRel** = complément-relative

TAB. 1 – Résultats de l'évaluation

3. les groupes prépositionnels (GP) : rappelons tout d'abord que les grammaires traditionnelles distinguent 2 types de compléments prépositionnels dépendants du verbe : le complément d'objet indirect (argument) et le complément circonstanciel (ajout) qui n'ont pas la même valeur sémantique. Toutefois, aucun des 2 analyseurs ne les distinguent l'un de l'autre, les considérant tous deux comme des modificateur du verbe. Cette différenciation, impliquant un calcul de cohésion entre le verbe et son complément, voir (Fabre & Frérot, 2002), n'est pas assurée par les analyseurs partiels. Nous avons cependant constitué 2 corpus de groupes prépositionnels verbaux permettant de faire apparaître cette distinction : V-GP1 contient des GP directement après le verbe (COI), V-GP2 présente des COI et des GP plus distants du verbe. On voit que les scores de LP, très élevés pour V-GP1, baissent avec l'apparition de GP ajouts dans V-GP2. De son côté, ACD a des scores moins bons mais égaux dans les deux catégories. Cette différence vient du fait que LP répertorie et distingue les verbes transitifs des intransitifs dans sa grammaire.

Les groupes prépositionnels ne dépendent pas toujours du verbe et peuvent modifier un nom (“*region of...*”) ou un adjectif (“*able to...*”). Se pose alors le problème du rattachement prépositionnel lorsque de tels groupes apparaissent après un verbe et son complément : dépendent-ils directement du verbe ou de son complément ? Sans connaissances lexicales ou sémantiques supplémentaires, de telles constructions relèvent souvent de l'ambiguïté. Deux possibilités de traitement sont envisagés : tandis qu'ACD fournit les 2 relations "V-GP" et "N-GP" (précision diminuée), LP est déterministe. LP opère un traitement spécifique des compléments en “*of*” en les rattachant systématiquement à l'objet. Ce choix est validé par l'amélioration du rappel et de la précision entre O-GP et NofN. Pour les NofN, comme le souligne l'écart entre les taux de précision des 2 analyseurs, le choix de la surgénération perd son intérêt éventuel. Le type d'exploitation des résultats de l'analyse peut toutefois influencer le choix entre exhaustivité et déterminisme.

Les groupes nominaux avec complément sont très courants dans notre corpus comme dans tout corpus de spécialité. Ils constituent souvent des termes et leur extraction est donc particulièrement intéressante et souhaitable. La difficulté, en dehors de la question de l'attachement, est que de tels groupes sont parfois complexes : il peuvent présenter des coordinations (N *of* (N *and* N)) ou des constructions imbriquées (N *of* (N *of* N)). Dans les deux cas, LP retrouve les 2 relations alors qu'ACD échoue sur les compléments en “*of*” contenant une coordination.

4. la coordination : c'est un phénomène très récurrent dans notre corpus et difficile à gérer, notamment pour les systèmes basés sur une grammaire de constituants. Il est effectivement difficile de prévoir le nombre de coordinations dans une phrase ou même le nombre de membres concernés par cette coordination. La coordination peut apparaître entre des noms, des verbes, des adjectifs, des adverbes ou des propositions, donc à différents niveaux syntaxiques. Lorsqu'il y en a plusieurs dans une même phrase, les possibilités augmentent de manière exponentielle. Le risque est alors de mélanger les groupes, voire les niveaux syntaxiques. De plus, l'analyseur doit prévoir la mise en relation de chaque élément de la coordination avec la tête dont il dépend (ce que ne fait pas ACD au niveau des relations). LP construit une nouvelle analyse complète de la phrase à chaque fois qu'il rencontre une coordination. La lisibilité des résultats en est améliorée mais le temps de traitement en subit les conséquences.

Chacun des deux analyseurs présente des comportements différents et plus ou moins adaptés face aux divers types de coordinations mais nous ne détaillerons pas les résultats. Notons simplement que Link Parser, de par le formalisme de sa grammaire, prévoit le traitement de nombreuses coordinations particulières (“*both...and*” ou deux verbes partageant le même auxiliaire ou le même

sujet). Les résultats, assez décevants pour les deux analyseurs (rappel < 0.8 pour LP, < 0.6 pour ACD), indiquent qu'il est nécessaire de continuer les recherches dans ce domaine et qu'il est intéressant de prendre la coordination en compte lors des évaluations. Les coordinations, sujet difficile, sont un bon indice du niveau de couverture des phénomènes complexes par un analyseur.

5. la négation : c'est aussi un problème difficile à gérer pour les analyseurs syntaxiques. La question est de savoir sur quel mot ou groupe elle porte. Or, son apparition dans les relations de dépendance est primordiale si on ne veut pas générer des informations fausses. Cet aspect peut être particulièrement crucial pour l'extraction d'information ou la traduction.

ACD ne gère les négations que dans les cas où elle font partie intégrante du groupe verbal ou adjectival. Elles apparaissent alors dans l'analyse syntagmatique, mais ne sont pas reportées dans les relations (rappel =0). LP les fait apparaître, mais uniquement lorsqu'elles portent sur le groupe verbal ("*be not*" ou "*do not V*") et qu'elles en sont proche. La relation entre la négation et le verbe est alors restituée et peut être exploitée. Lorsque la négation est éloignée de la tête du groupe (par une coordination par exemple), elle n'est analysée par aucun des deux systèmes. Comme pour la coordination, il semblerait que des travaux supplémentaires soient nécessaires pour améliorer le traitement de la négation en général.

6. les adjectifs comparatifs : les constructions comparatives sont assez courantes dans les textes rendant compte de mesures comme les résumés scientifiques. Il était donc intéressant pour nous d'avoir un système qui analyse ces constructions particulières, c'est pourquoi nous avons considéré dans nos tests les 2 constructions "... *more/less Adj/Adv than ...*" et "... *Adj-er than ...*". La difficulté réside dans le fait qu'elles apparaissent dans des contextes très divers : comparaison de noms, de groupes prépositionnels (PP), d'adjectifs, etc.

ACD n'attache l'adjectif pivot de la comparaison qu'au premier membre de la comparaison. Le second n'est attaché à rien, car "*than*" n'est en fait pas réellement analysé. LP réussit généralement bien l'analyse de telles constructions qui sont prévues par la grammaire, sauf lorsqu'il y a plusieurs comparatifs en "*more*" qui sont coordonnés. Après avoir retravaillé les règles de la grammaire de LP concernant les comparatifs (les scores sont un peu faibles pour "*more ... than*"), on peut envisager un post-traitement spécifique pour ces constructions aboutissant à l'extraction de relations exprimant la comparaison entre deux objets.

7. le passif : nous regroupons ici les passifs simples et les passifs apparaissant dans les relatives. Le passif est aujourd'hui relativement bien traité par les analyseurs car il soulève des questions importantes de l'analyse syntaxique (comme les formes profonde et de surface) et est depuis longtemps un thème de recherche en linguistique. Les résultats des deux analyseurs sont satisfaisants, bien qu'un peu meilleurs pour LP, surtout dans des propositions relatives.

Les patients (sujet du passif) sont en général restitués correctement par les 2 systèmes, particulièrement dans le cas des passifs simples. Les agents, compléments de forme "*by X*", sont bien reconnus mais ACD est encore une fois pénalisé par la surgénération des relations (compléments en "*by*" identifiés comme agents du verbe au passif et comme modificateurs du verbe). Ceci explique des taux de précision plus faibles pour ACD.

8. les propositions relatives (relatives dont l'antécédent est sujet et celles dont il est complément et relatives au passif) : les relatives sont très nombreuses dans notre corpus et fournissent des relations sujet-verbe et verbe-objet (non explicites), qui nous intéressent particulièrement.

Les difficultés sont multiples dans l'analyse de telles constructions. La première est de bien délimiter les propositions, principale et relative, à l'aide notamment du pronom relatif. Il faut noter que "*that*" pose parfois des problèmes à cause de sa double nature, déterminant et pronom relatif. La présence d'un verbe au passif dans la principale ajoute une difficulté supplémentaire qu'ACD n'arrive pas toujours à surmonter. Une fois les verbes des propositions repérés, leurs sujets et objets respectifs doivent être retrouvés. Les deux analyseurs ne montrent pas de fiabilité particulière au niveau de cette analyse locale.

La dernière étape est la mise en relation du pronom avec son antécédent (qui se trouve dans la principale). Que l'antécédent soit sujet ou complément dans la principale, le sujet réel du verbe de la relative (SuRel) est correctement retrouvé par les 2 systèmes. Lorsqu'il est complément réel de la relative (CoRel), ACD ne restitue pas l'antécédent et présente donc des scores très mauvais. LP est capable de traiter plus de constructions complexes ou spécifiques qu'ACD car sa grammaire permet des descriptions plus fines et plus spécifiques de la langue.

Conclusion : Le tableau 1 montre que Link Parser obtient généralement des meilleurs scores qu'ACD. La principale raison d'un tel écart est que nous avons axé notre évaluation sur des constructions et relations complexes. Nous aurions pu considérer seulement les relations sujet-verbe, verbe-objet, nom-prep-nom et verbe-prep-nom ainsi que quelques relations simples (car mettant en oeuvre des éléments proches) comme adjetif-nom, nom-nom ou encore adverbe-verbe. Les résultats que nous aurions alors obtenus auraient sans aucun doute été meilleurs et plus proches entre les deux analyseurs. Ce que nous voulions évaluer en premier lieu était la robustesse du système que nous devions choisir. Il devait être capable de fournir le plus d'informations correctes possibles pour tout type de phrase. Notre corpus étant principalement composé de phrases très longues et de complexité élevée, voire très élevée, notre hypothèse était qu'un analyseur capable de retrouver des relations difficiles pourrait évidemment traiter des relations relativement triviales.

Il se trouve que Link Parser a obtenu des résultats dépassant nos espérances. Ceci est probablement dû à son dictionnaire-grammaire qui permet de constituer des familles de mots ayant un même comportement syntaxique. Plus de précision est donc possible sans création d'ambiguités. De façon similaire, il est possible de décrire le comportement de mots, expressions ou constructions très spécifiques. L'adoption de Link Parser nous offrait donc la perspective d'adapter l'analyse à notre domaine d'étude, à savoir les articles scientifiques et plus précisément ceux ayant trait à la génomique fonctionnelle. Une adaptation de LP à la langue de spécialité nous a ensuite permis d'améliorer l'analyse pour laquelle nous évaluons prochainement les scores de précision et de rappel.

La seconde raison qui explique les meilleurs résultats de Link Parser par rapport à ceux d'ACD est que ce sont les relations syntaxiques que nous avons évaluées et non le découpage en syntagmes des phrases. Link Parser étant basé sur une grammaire de dépendances, les relations syntaxiques sont pour lui à la fois la fin et les moyens de l'analyse, ce qui en fait un outil particulièrement bien adapté à notre évaluation. Dans ACD, les relations syntaxiques ne font que découler de l'analyse en syntagmes. De plus, avec ce dernier, elles apparaissent explicitement dans le résultat alors que LP ne fournit en sortie que des relations directes qu'il est nécessaire d'associer par la suite pour retrouver les paires de mots qui nous intéressent, voir 2.3. La reconstitution des relations complexes n'étant pas un exercice trivial, il faut être prudent dans nos conclusions et reconnaître que Link Parser n'accomplit pas cette dernière étape du traitement qui présente des difficultés non négligeable. Toutefois, Link Parser restitue un très grand nombre d'informations et il ne tient qu'à nous de les exploiter judicieusement par la suite. Il

semble enfin qu'ACD ne construise pas tous les types de relations, ne fournissant qu'un résultat partiel, contrairement à LP.

5 Conclusion

La première information que nous tirons de cette évaluation est la meilleure performance de Link Parser face à ACD dans le traitement de phrases issues d'un corpus de spécialité. Nous pouvons donc avancer l'hypothèse que les grammaires basées sur les dépendances sont particulièrement bien adaptées au traitement de phrases longues et complexes.

De plus, notre méthode d'évaluation organisée autour de phénomènes linguistiques particuliers nous a permis d'identifier les difficultés rencontrées par Link Parser et d'imaginer des solutions pour améliorer l'analyse avant même son intégration dans notre système d'extraction d'information.

Nous insistons sur l'intérêt de développer les évaluations basées sur les relations syntaxiques puisque ces dernières sont très riches en information et particulièrement utiles dans certaines applications demandant une analyse syntaxique comme les systèmes d'extraction d'information ou de traduction. La croissance du nombre des analyseurs en dépendances ou hybrides ces dernières années doit être une motivation supplémentaire.

Références

- BLACK E., ABNEY S., FLICKENGER D., GDANIEC C., GRISHMAN R., HARRISON P., HINDLE D., INGRIA R., JELINEK F., KLAVANS J., LIBERMAN M., MARCUS M., ROUKOS S., SANTORINI B. & STRZALKOWSKI T. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In DARPA, Ed., *Proceedings of the fourth DARPA Speech and Natural Language Workshop*, p. 306–311.
- FABRE C. & FRÉROT C. (2002). Groupes prépositionnels arguments ou circonstants : vers un repérage automatique en corpus. In *actes de TALN, Nancy*, p. pp. 215–224.
- FAURE D. & NÉDELLEC C. (1999). Knowledge acquisition of predicate argument structures from technical texts using Machine Learning : the system ASIUM. In D. F. R. STUDER, Ed., *11th European Workshop EKAW'99*, p. 329–334.
- GENDNER V., ILLOUZ G., JARDINO M., MONCEAUX L., PAROUBEK P., ROBBA I. & VILNAT A. (2002). A Protocol for Evaluating Analysers of Syntax (PEAS). In *LREC 2002, Espagne*.
- LIN D. (1998). Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*.
- PRASAD R. & SARKAR A. (2000). Comparing Test-suite based evaluation and Corpus-based evalution of a wide-coverage grammar for English. In *Using Evaluation within Human Language Technology Programs : Results and Trends. LREC'2000 Satellite Workshop*.
- SCHNEIDER G. (1998). *A linguistic Comparison of Constituency, Dependency and Link Grammar*. PhD thesis, Institut für Informatik der Universität Zürich, Suisse.
- SLEATOR D. & TEMPERLEY D. (1991). *Parsing English with a Link Grammar*. Rapport interne, Carnegie Mellon University.
- SRINIVAS B., SARKAR A., DORAN C. & HOCKEY B. (1998). Grammar and Parser Evaluation in the XTAG Project. In *Workshop on the Evaluation of Parsing Systems*.
- TEMPERLEY D. (1999). An Introduction to the Link Grammar Parser. Carnegie Mellon University.

Une grille d'évaluation pour les analyseurs syntaxiques

Philippe Blache (1) & Jean-Yves Morin (2)

(1) LPL-CNRS, Université de Provence

29, Avenue Robert Schuman

13621 Aix-en-Provence

pb@lpl.univ-aix.fr

(2) Département de Linguistique, Université de Montréal

CP 6128, succ. Centre-ville

Montréal QC H3C 3J7

morinjy@sympatico.ca

Mots-clefs – Keywords

Evaluation, grille d'analyse, performance, diagnostic

Evaluation template, performance, diagnostic

Résumé - Abstract

Les techniques d'évaluation aujourd'hui disponibles posent un certain nombre de problèmes à la fois pour ce qui concerne la disponibilité des ressources nécessaires, mais également dans la mesure où elles ne reflètent pas complètement des véritables capacités d'un système. Nous proposons dans cet article l'élaboration d'une grille d'analyse qui constitue une description standardisée des caractéristiques du système. Cette grille contient à la fois des aspects purement descriptifs (concernant par exemple le formalisme ou les aspects algorithmiques) ainsi qu'un ensemble de mesures automatisées, le tout donnant une image plus précise du système qu'une simple évaluation quantitative.

The evaluation procedures that have been tested especially for english or german pose some problems. First, resources required for a quantitative evaluation (typically a treebank) are not well developped for french. Moreover, such measures only concern subset of the system capabilities. We propose in this paper an evaluation template containing both descriptive characteristics (presentation of the formalism, faithfulness of the implementation, algorithmic aspects) and automatic measurements (recall, precision, relations, tests suites, etc.).

1 Introduction

Les résultats des précédentes campagnes d'évaluation, en particulier pour ce qui concerne les méthodologies mises en œuvre, doivent être analysée de près avant de s'engager dans le développement d'un nouveau protocole d'évaluation en France. De ce point de vue, il est intéressant de revenir d'une part sur les campagnes engagées par l'AUPELF dans le milieu des années 90 (cf. (9) dont les résultats ont joué un rôle majeur dans la création de la conférence

LREC, ainsi bien entendu que sur les campagnes d'évaluations proposées pour l'anglais et en particulier sur les résultats de Parseval (cf. (14)).

Sans entrer dans le détail de tous les projets, il convient, à titre de remarque préliminaire, de constater qu'un même modèle d'évaluation ne peut convenir à n'importe quel système de traitement de la langue. Les systèmes de traitement de la parole étaient par exemple dans les années 90 bien plus avancés que les systèmes de traitement de l'écrit et pouvaient se prêter plus facilement à des évaluations quantitatives. Pour ce qui concerne l'écrit, s'il était possible d'évaluer et de comparer de façon assez précise des systèmes d'étiquetage morpho-syntaxiques (cf. campagne GRACE), il était en revanche beaucoup plus difficile (pour ne pas dire impossible) de dresser ne serait-ce qu'un véritable protocole pour l'évaluation des systèmes de compréhension (cf. (2)). L'évaluation des analyseurs syntaxiques n'échappe pas à cette règle et les activités menées dans le cadre de Parseval ont montré les mêmes limites. Il n'est pas possible de réduire l'évaluation de tels systèmes à une simple évaluation quantitative reposant sur la détection de frontières et de types d'unités. Encore faut-il également savoir de quelles unités parle-t-on ainsi que du niveau d'information nécessaire à l'analyse. Ainsi, un analyseur s'appuyant exclusivement sur les informations syntaxiques sans autre type d'information (par exemple la sémantique lexicale), devra donner comme résultat l'ensemble des structures syntaxiques possibles. Il s'agit donc d'un premier problème ayant des conséquences importantes sur les ressources utilisées pour l'évaluation : un corpus de référence annoté sur la base duquel sont évalués les résultats devra par exemple comporter cet ensemble d'analyses. Le problème se complique encore si on cherche à comparer des analyseurs. Il existe en effet un grand nombre de formalismes linguistiques et il n'est pas possible de concevoir une représentation générique de l'information syntaxique qui serait utilisable quel que soit le formalisme. Enfin, et cette question relève tout autant du génie logiciel, il est indispensable de connaître le niveau de généralité du système, ainsi que ses capacités d'évolution et d'adaptation.

Nous allons dans cet article revenir sur l'ensemble de ces points. Il est clair qu'une campagne d'évaluation doit prendre en compte un certain nombre de critères dépassant largement la stricte évaluation quantitative. Mais il est dans le même temps indispensable de s'interroger non seulement sur les ressources nécessaires à une telle évaluation ainsi que sur notre capacité à les développer. Plus généralement, encore, la question préliminaire, avant de savoir comment évaluer des systèmes, est de savoir pourquoi évaluer ces systèmes et qu'entend-on par évaluation. La réponse à ces questions permet de mettre en perspective l'utilisation de techniques et ressources existantes tout en les situant dans un cadre plus général. En d'autres termes, nous proposons une approche permettant de réutiliser les protocoles existants en offrant une possibilité d'interprétation dans un cadre générique.

2 Situation

Nous nous proposons dans cette section de faire le point sur les différentes techniques ou propositions faites dans le domaine de l'évaluation.

Les critères d'évaluation des analyseurs syntaxiques et plus généralement des systèmes de traitement des langues naturelles (cf. notamment (10), (22) ou (2)) sont de plusieurs types. Ils portent bien entendu sur des mesures quantitatives, mais également sur l'adéquation du système aux objectifs visés ainsi qu'à sa capacité d'évolution. Nous décrivons plus particulièrement dans cette section les principales approches portant sur le comportement et les résultats fournis par un système.

Les critères classiques de l'évaluation d'applications, typiquement le rappel et la précision, permettent de situer les résultats par rapport à un standard. La quantification de ces critères se fait donc sur la base d'un corpus annoté, typiquement un corpus arboré du type Penn treebank (cf. (20)) ou Susanne (cf. (21)). On parle dans ce cas d'évaluation de la performance d'un système. On utilise pour cela des procédures entièrement automatisées.

Mais il existe également un second type d'évaluation qui relève plutôt du diagnostic. Il s'agit dans ce cas d'examiner les résultats fournis par un système pour l'analyse d'un ensemble de phrases tests comportant des difficultés ou des tournures particulières, y compris non grammaticales.

2.1 Evaluation sur corpus arborés

Ce type de corpus arboré propose essentiellement l'annotation d'informations portant sur le type des catégories, leur étendue ainsi que d'autres informations complémentaires (mais de façon moins systématique) concernant par exemple la structure prédicative. Il s'agit donc typiquement de parenthésage encodant essentiellement des informations sur la constituance ainsi que quelques relations syntaxiques à proprement parler. Les structures proposées dans ce type de corpus sont plates plutôt que profondes, et le nombre d'emboîtements est limité. Ce choix est essentiellement fonctionnel et favorise généralement le consensus entre annotateurs. Cette caractéristique implique l'utilisation de catégories de niveau relativement haut de préférence à des catégories de granularité plus fine (cf (3)). Par ailleurs, les informations encodées relèvent exclusivement du niveau syntaxique. Enfin, une structure syntaxique est choisie pour être encodée dans le cas d'analyses multiples, au détriment de la représentation de l'ambiguïté. Ce premier type de corpus, de loin le plus répandu pour l'anglais, repose donc sur un certain nombre de choix concernant à la fois le formalisme linguistique ainsi que le type d'information à encoder.

La première technique d'évaluation consiste simplement à mesurer dans quelle mesure un analyseur est capable de reproduire les informations contenues dans ce type de corpus. C'est typiquement le cas des évaluations menées dans le cadre de Parseval qui propose de mesurer la précision, le rappel ainsi que le nombre de croisements de parenthèses entre le corpus de référence et la sortie de l'analyseur.

Plusieurs critiques sont faites à ce type d'évaluation. Tout d'abord, et c'est la critique principale, les mesures proposées sont valides pour des analyseurs syntagmatiques. Un certain nombre d'annotations notamment du Penn treebank relèvent encore plus précisément de choix théoriques et formels pertinents dans un cadre particulier, mais peu adaptés pour d'autres approches. Par ailleurs le choix des étiquettes associées aux catégories, en d'autre termes le typage des objets manipulés, est d'une granularité élevée, ce qui ne permet pas toujours la description de phénomènes variés. Plusieurs auteurs (cf. par exemple (18)) ont montré que ce type de mesures pénalise les analyseurs proposant des annotations plus précises (donc un plus grand nombre de parenthèses que dans le corpus de référence).

2.2 Evaluation sur des suites de phrases

Plusieurs projets ou campagnes d'évaluation ont proposé de tester les performances des systèmes sur la base d'un ensemble de phrases tests. La plupart des grands projets de développement d'analyseurs syntaxiques ont élaboré leur propre jeu de phrases. C'est le cas par exemple du projet Alvey (cf. (6)) proposant des analyseurs basés sur GPSG ou d'un projet similaire mené

par le laboratoire Hewlett-Packard (cf. (12)). On doit également signaler en France une première opération de comparaison de plusieurs analyseurs pour le français proposant également un tel jeu (cf. (11)).

Mais l'opération la plus systématique en termes de ressources développées, de couverture et de nombre de langues reste bien entendu TSNLP (cf. (16) et (17)). Ce projet a proposé une méthodologie, des conventions d'annotation ainsi que des outils pour l'élaboration de tels jeux. L'idée est d'identifier un certain nombre de phénomènes linguistiques et pour chacun d'entre eux, de proposer une série de phrases. Une des particularités de TSNLP est de proposer également des phrases mal formées, ce qui permet de tester le comportement de l'analyseur en terme de reconnaissance mais également de robustesse.

Parmi les phénomènes listés, notamment pour le français, nous trouvons la complémentation, la modification, l'accord, la coordination, la négation, etc. Ce type d'information est utile pour une évaluation dite *diagnostique* du système mais également indispensable pour analyser l'évolution du développement de la grammaire et de l'analyseur.

2.3 Evaluations relationnelles

Un des problèmes majeurs posés à l'évaluation vient du fait que, à la différence des étiqueteurs morphologiques, l'analyse syntaxique repose sur des formalismes variés. Il est donc impossible de prétendre représenter des structures syntaxiques complètes sans être dépendant d'un formalisme donné. Ainsi, la plupart des corpus annotés s'appuient sur des grammaires syntagmatiques. Il n'est donc pas possible d'utiliser *directement* ces corpus pour évaluer des analyseurs utilisant d'autres formalismes comme les grammaires de dépendance par exemple.

Dans la mesure où il n'est pas concevable ni d'un point de vue matériel ni même d'un point de vue d'efficacité de développer des ressources adaptées à des formalismes, il convient de chercher une démarche intermédiaire. Une solution proposée consiste à exploiter pour l'évaluation non pas les structures syntaxiques (en particulier les unités et leurs frontières), mais plutôt les relations syntaxiques. On trouve des propositions allant dans ce sens dans (15) ou encore (19) qui suggère d'utiliser des relations de dépendances plutôt que de constituance y compris pour évaluer des analyseurs syntagmatiques. Dans cette perspective, chaque mot est associé à trois types d'informations :

- la catégorie du mot courant
- la tête modifiée par le mot courant et sa localisation par rapport à celui-ci
- le type de relation qui unit les deux mots : sujet, adjoint, complément, spécifieur, etc.

Une autre approche plus systématique et encore plus générale propose de s'appuyer sur les relations grammaticales plutôt que sur d'autres types d'informations. Il s'agit du schéma d'annotation de relations grammaticales (cf. (7), (8)). Les auteurs proposent l'annotation d'un ensemble de relations syntaxiques identifiables indépendamment du formalisme choisi. De plus, ces relations sont hiérarchisées ce qui permet des niveaux de précision différents dans l'annotation. Nous rappelons brièvement le type de relations proposées par ces auteurs.

Une grille d'évaluation pour les analyseurs syntaxiques

Niveau	Nom	Arguments	Description
1	<i>dependent</i>	introduction tête dépendant	Relation de dépendance générique entre une tête et un dépendant
1.1	<i>mod</i>	type tête dépendant	Relation entre une tête et son modifieur. Le type est le mot introduisant la dépendance
1.1.1	<i>ncmod</i>	-	Modificateur lexical (non clausal)
1.1.2	<i>xmod, cmod</i>	-	Modificateurs propositionnels
1.2	<i>arg-mod</i>	type tête dépendant <i>rel-initiale</i>	Relation tête/argument, celui-ci étant réalisé comme un modifieur
1.3	<i>arg</i>	tête dépendant	Relation générique tête/argument (plutôt de type complément)
1.3.1	<i>subj</i>	tête dépendant relation	Relation prédicat/sujet
1.3.1.1	<i>ncsubj</i>	-	Sujet lexical (non clausal)
1.3.1.2	<i>xsubj, csubj</i>	-	Sujets propositionnels
1.3.2	<i>comp</i>	tête dépendant	Relation tête/complément
1.3.2.1	<i>obj</i>	-	Relation tête/objet
1.3.2.1.1	<i>dobj</i>	-	Relation prédicat/objet direct (premier complément non propositionnel)
1.3.2.1.2	<i>iobj</i>	-	Relation prédicat/complément non propositionnel introduit par une préposition
1.3.2.1.3	<i>obj2</i>	-	Relation prédicat/second complément non propositionnel
1.3.2.2	<i>clausal</i>	-	Relation tête/complément propositionnel

Il est important de rappeler pour terminer, si besoin était, que le type de ressource que nous venons de décrire n'existe que marginalement pour le français. Une seule véritable ressource a été développée sous l'impulsion de Anne Abeillé qui a, avec son équipe, produit un corpus arboré sur la base de textes issus du journal "Le Monde". L'exemple suivant donne un extrait de cette ressource pour la phrase "*Seuls pirates, marchands d'esclaves et trafiquants de drogue y sont légitimement poursuivis par tous*" (les balises fermantes sont omises pour des raisons de lisibilité).

```

<SENT nb="10000">
  <NP>
    <w lemma="seul" ei="AImp" ee="A-ind-mp" cat="A" subcat="ind" mph="mp">Seuls</w>
    <w lemma="pirate" ei="NCmp" ee="N-C-mp" cat="N" subcat="C" mph="mp">pirates</w>
  <COORD>
    <w lemma="," ei="PONCTW" ee="PONCT-W" cat="PONCT" subcat="W"><,>

```

3 Quelques informations de base pour la grille

3.1 Evaluation des aspects syntagmatique

L’analyse de la situation décrite précédemment nous permet de dégager quelques directions de réflexion. Les expériences montrent tout d’abord qu’il ne faut pas s’engager dans l’annotation d’informations syntaxiques en utilisant un formalisme trop spécifique pas plus qu’on ne peut prétendre à proposer une annotation générique. Mais plusieurs approches ont montré qu’il était possible d’extraire un certain type d’information, plutôt de niveau relationnel, à partir d’encodages variés.

Par ailleurs, les travaux récents ont montré les limites d’une seule évaluation des performances sur la base d’un parenthésage. Ce type d’information, en plus du fait qu’il est dépendant d’un formalisme, pose de nombreux problèmes, notamment pour une évaluation précise. Mais là encore, il ne faut pas négliger le fait que la plupart des analyseurs superficiels (donc une bonne proportion des systèmes d’analyse aujourd’hui disponibles) produisent de telles structures. Il est donc nécessaire de préserver la possibilité d’exploiter ces informations. Il faut cependant compléter ces aspects par une granularité plus fine des mesures. Imaginons par exemple que nous ayons 85% de SN de la forme Dét+N. Le parenthésage et l’étiquetage incorrect des SN qui ne sont pas de ce type devrait donc tenir compte de cet aspect et l’évaluation devrait être pondérée en fonction de la fréquence et/ou de la complexité de la structure.

A ce stade, il est nécessaire d’aborder le problème de la distinction analyseur/grammaire. En effet, tous les protocoles évaluent aujourd’hui conjointement ces deux composants. Mais il est intéressant et sans doute important d’entrer dans un niveau de description plus fin, y compris en termes computationnels. Les analyseurs symboliques distinguent les parties données et traitement. Une grammaire est dans ce cas clairement distincte de l’analyseur qui l’utilise. Même si une telle distinction n’est pas valide pour d’autres techniques, il faut pouvoir distinguer dans l’évaluation la qualité de la grammaire de celle de l’analyseur à proprement parler. Il s’agit bien entendu d’une entreprise extrêmement difficile que de tenter de caractériser ces deux aspects, souvent indissociables. On peut toutefois citer un certain nombre de caractéristiques propres à l’analyseur. Il est important par exemple de juger du comportement du système pour le traitement des non attendus, sa robustesse. Il est également important de prendre en compte le déterminisme de l’analyse et la capacité à hiérarchiser les solutions en cas de non-déterminisme. Sans entrer dans une tentative de quantification de ces aspects, un élément d’information intéressant réside dans le type et la quantité de structures intermédiaires produites en cours d’analyse (par exemple, pour un analyseur tabulaire, le nombre total d’arcs utilisés pour une analyse donnée). Pour ce qui concerne la grammaire, l’évaluation distincte est difficile à cerner en dehors d’une comparaison entre plusieurs versions d’une grammaire. D’un point de vue purement descriptif, il est malgré tout important de connaître le nombre et le type de catégories utilisées, le niveau de hiérarchisation (s’il existe) ou le type d’encodage de l’information (règles, contraintes, etc.).

Le dernier aspect qui nous semble déterminant à prendre en compte concerne le formalisme lui-même. Il est en effet, nous y reviendrons dans la section suivante, important de connaître le paradigme théorique dans lequel se situe l’analyseur et surtout, puisqu’un des aspects de l’évaluation est la comparaison entre plusieurs systèmes, de connaître le degré de fidélité de l’implantation à cette théorie. Nous sommes ainsi en mesure de savoir, dans le cas où une théorie aurait un intérêt particulier au-delà de la syntaxe par exemple, s’il est possible d’exploiter la

totalité de son pouvoir expressif ou pas.

3.2 Informations non syntagmatiques

Les informations purement syntagmatiques sont bien entendu fondamentales, mais il est également très important d'évaluer les autres dimensions comme les fonctions grammaticales, les rôles abstraits, la structure communicative, la structure anaphorique, la deixis, etc. Là encore, il ne s'agit pas de limiter l'évaluation aux simples procédures de mesures quantitatives. Même si nous ne disposons pas de corpus de référence encodant ce type d'information, il est important de disposer d'une description précise de la prise en compte par le système de ces informations.

Par ailleurs, la syntaxe entretient des liens étroits avec d'autres domaines linguistiques. Il est donc nécessaire de prendre en compte la capacité de l'approche et du système en particulier à représenter et traiter les relations avec d'autres domaines proches comme la morphologie ou la sémantique. Cette question d'interfaçage est déterminante pour plusieurs raisons. D'une part, elle fournit une indication sur les potentialités d'utilisation du système. D'autre part, un analyseur ouvert sur d'autres domaines est révélateur d'une technologie plus durable et à terme plus efficace. A un niveau plus général, ces paramètres nous semblent concourir à une évaluation, ou du moins une spécification, des capacités d'évolution du système.

4 Une grille d'évaluation des analyseurs

Avant de décrire plus précisément la grille d'évaluation que nous proposons, il est important de tirer quelques conclusions des remarques faites précédemment. Tout d'abord, face à la difficulté de la tâche de constitution de ressources pour l'évaluation (en particulier les corpus annotés), il semble nécessaire de tirer parti des toutes les ressources disponibles, quelles qu'elles soient. Par ailleurs, il semble tout aussi nécessaire de ne pas limiter l'évaluation aux simples mesures de performance de l'analyseur. Il convient de situer cette remarque dans la perspective d'une question plus générale : à quoi sert l'évaluation de tels systèmes ? La réponse varie en fonction de la destination : elle sert de validation pour le développeur lui-même, mais elle est aussi indicatrice des capacités du système pour un utilisateur potentiel qui chercherait le système le plus adapté à ses besoins. Il faut donc compléter les mesures ou benchmarks par une description du système à proprement parler. Nous proposons donc de distinguer deux types d'informations dans la grille d'évaluation :

- les évaluations quantitatives : ensemble d'opérations automatiques ou semi automatiques mesurant des résultats d'analyse sur des corpus de référence annotés ou des ensembles de phrases-tests. Une description analytique des sorties pourra compléter les mesures (typiquement le comportement du système sur des entrées mal formées),
- la description du système : informations fournies par le concepteur du système concernant les aspects non mesurables automatiquement.

On récapitule dans le tableau suivant l'ensemble des caractéristiques évoquées précédemment et qui nous semblent devoir entrer en ligne de compte pour une évaluation précise, voire une comparaison des systèmes d'analyse syntaxique. Il ne s'agit pas bien entendu d'une liste exhaustive.

De même en fonction des objectifs de l'évaluation, il peut ne pas être utile de renseigner tous les champs de la grille. Il nous semble cependant utile de fournir une vision aussi précise du système qui permette de mettre en perspective les mesures quantitatives par rapport à la base théorique et computationnelle du système.

Description	<ul style="list-style-type: none"> • Formalisme, théorie <ul style="list-style-type: none"> – Description du cadre théorique – Description de la fidélité de l'implantation à ce cadre théorique, analyse des simplifications si elles existent. • Description du type d'information retourné par le système <ul style="list-style-type: none"> – Catégories : types de catégorie, granularité, représentation – Représentation de l'information syntaxique : règles, relations, etc. – Structures construites : arbres, graphes, ensembles, etc. • Ressources utilisées : <ul style="list-style-type: none"> – Lexiques, grammaires – Outils • Description algorithmique <ul style="list-style-type: none"> – Architecture – Stratégie d'analyse, déterminisme – Hiérarchisation des solutions – Complexité
-------------	---

Le tableau présenté ci-après récapitule l'ensemble des mesures qu'il nous semble intéressant de faire. Là encore, il ne s'agit pas d'une liste exhaustive prétendant évaluer précisément tous les aspects du système. Il nous semble cependant important, compte tenu de toutes les critiques faites aux autres protocoles, de proposer une approche mixte, tirant parti de toutes les techniques existantes dans ce domaine. Une telle démarche permet de plus de tirer le meilleur parti des ressources existantes sans les mettre en concurrence.

Mesures	<ul style="list-style-type: none"> • Parenthésage <ul style="list-style-type: none"> – Description du corpus de référence – Rappel – Précision – Croisements • Relations <ul style="list-style-type: none"> – Définition de l'ensemble des relations à évaluer – Description du corpus de référence – Quantification • Phrases tests <ul style="list-style-type: none"> – Définition de l'ensemble des tournures visées – Description du jeu de phrases – Quantification – Description des analyses pour les entrées mal formées
---------	---

La partie d'évaluation des relations mérite quelques commentaires. Il s'agit là de s'inscrire dans la démarche proposée par (8) et donc de spécifier une ensemble de relations syntaxiques nous semblant refléter à la fois de la complexité du problème ainsi que des capacités au moins partielle des systèmes. La liste des relations n'est pas figée. La proposition récapitulée dans la section précédente constitue une base de départ qui peut éventuellement être complétée voire adaptée pour le français (certaines relations peuvent en effet être pertinentes essentiellement pour l'anglais).

5 Conclusion

Les techniques d'évaluation aujourd'hui disponibles posent un certain nombre de problèmes. Tout d'abord, une évaluation purement quantitative ne permet pas de caractériser précisément les capacités d'un système. De plus, la disponibilité des ressources nécessaires pour une telle évaluation est contrastée selon les langues. Avant de s'engager dans une campagne d'évaluation plus systématique des analyseurs existants pour le français, il nous a donc semblé utile de faire le point de la situation et de proposer, plutôt qu'une métrique ou un véritable protocole, un cadre général que nous appelons grille d'évaluation. L'idée est d'une part de rendre compte d'aspects généraux caractérisant l'analyseur (par exemple le formalisme choisi ou encore certaines caractéristiques algorithmiques) et d'autre part de rassembler plusieurs techniques d'évaluation (diagnostic, performances, etc.). Une telle approche nous semble être raisonnable dans la mesure où, plutôt que d'élaborer une théorie de l'évaluation, nous rassemblons plusieurs indices ou critères caractérisant au mieux les systèmes.

Références

1. [Atwell96] Atwell E. (1996) "Comparative evaluation of grammatical annotation models" In R. Sutcliffe, H. Koch, A. McElligott (Eds.), *Industrial Parsing of Software Manuals*, Rodopi.
2. [Blache97] Blache P., J. Guizol, F. Lévy, A. Nazarenko, S. N'Guema, M. Rolbert, R. Pasero & P. Sabatier (1997) "Evaluer des systèmes de compréhension de textes", in Actes des Journées Scientifiques et Techniques (JST 97, Avignon), AUPELF-UREF.
3. [Blache98] Blache P. (1998) "A quoi sert l'annotation syntaxique de corpus ?", in *Corpus. Méthodologie et applications linguistiques*, M. Bilger (ed.), Champion.
4. [Briscoe95] Briscoe, E., Carroll, J. (1995) "Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels", in Proceedings of *IWPT'95*.
5. [Carpenter97] Carpenter B. & Manning, C. (1997) "Probabilistic parsing using left corner language models", in Proceedings of the *5th ACL/SIGPARSE International Workshop on Parsing Technologies*.
6. [Carroll91] Carroll, J., E. Briscoe & C. Grover (1991) "A development environment for large natural language grammars", Computer Laboratory, Cambridge University, UK, Technical Report 233.
7. [Carroll98] Carroll, J., Briscoe E. & Sanfilippo, A. (1998) "Parser evaluation: a survey and a new proposal", in Proceedings of the *International Conference on Language Resources and Evaluation*.
8. [Carroll02] Carroll J., G. Minnen & T. Briscoe (2002) "Parser evaluation: using a grammatical relation annotation scheme", in A. Abeillé (ed), *Trebanks: Building and Using Syntactically Annotated Corpora*, Kluwer.
9. [Chibout00] Chibout K., F. Néel, J. Mariani et N. Masson (eds) (2000) *Ressources et Evaluation en Ingénierie de la Langue*. Duculot / De Boeck-Université.
10. [Cole96] Cole R., J. Mariani, H. Uszkoreit, A. Zaenen & V. Zue eds. (1996) *Survey of the state of the art in human language technology*, <http://www.cse.ogi.edu/CSLU/HLTsurvey/>

11. [Fay-Varnier91] Fay-Varnier C., C. Fouqueré, G. Prigent, & P. Zweigenbaum (1991) “Modules syntaxiques des systèmes d’analyse du français”, in *Technique et Science Informatiques*, 10(6).
12. [Flickinger87] Flickinger D., J. Nerbonne, I. Sag & T. Wasow (1987) “Toward Evaluation of NLP Systems”, HP Labs Technical Report, Palo Alto.
13. [Gaizauskas98] Gaizauskas, R., Hepple M., Huyck, C. (1998) “Modifying existing annotated corpora for general comparative evaluation of parsing”, in Proceedings of the *LRE Workshop on Evaluation of Parsing Systems*.
14. [Harrison91] Harrison, P., Abney, S., Black, E., Flickinger, D., Gdaniec, C., Grishman, R., Hindle, D., Ingria, B., Marcus, M., Santorini, B., Strzalkowski, T. (1991) “Evaluating syntax performance of parser/grammars of English”, in Proceedings of the *Workshop on Evaluating Natural Language Processing Systems*, 29th Annual Meeting of the Association for Computational Linguistics.
15. [Kübler02] Kübler S. & H. Telljohann (2002) “Towards a Dependency-Oriented Evaluation”, in Proceedings of *Beyond Parseval Workshop*.
16. [Lehmann96a] Lehmann, S., Oepen, S., Regnier-Prost, S., Netter, K., Lux, V., Klein, J., Falkedal, K., Fouvry, F., Estival, D., Dauphin, E., Compagnon, H., Baur, J., Balkan, L., Arnold, D. (1996) “TSNLP - test suites for natural language processing”, in Proceedings of *COLING’96*.
17. [Lehmann96b] Lehmann S., D. Estival & S. Oepen (1996) “TSNLP - test suites for natural language processing”, in Proceedings of *COLING’96*.
18. [Lin98] Lin, D. (1998) “A dependency-based method for evaluating broad-coverage parsers”, in *Natural Language Engineering*.
19. [Lin02] Lin, D. (2002) “Dependency-based evaluation of MINIPAR”, in A. Abeillé (ed), *Trebanks: Building and Using Syntactically Annotated Corpora*, Kluwer.
20. [Marcus93] Marcus, M., Santorini, B., Marcinkiewicz, M. (1993) “Building a large annotated corpus of English: The Penn Treebank”, in *Computational Linguistics*, 19(2).
21. [Sampson95] Sampson G. (1995) *English for the Computer: The SUSANNE Corpus and Analytic Scheme* Oxford University Press.
22. [Sparck Jones96] Sparck Jones K. & J. Galliers (1996) *Evaluating Natural Language Processing Systems*, Springer.
23. [Srinivas96] Srinivas, B., Doran, C., Hockey B., Joshi A. (1996) “An approach to robust partial parsing and evaluation metrics”, in Proceedings of the *ESSLI’96 Workshop on Robust Parsing*.

Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS

V. Gendner, G. Illouz, M. Jardino,
P. Paroubek, L. Monceaux, I. Robba, A. Vilnat

LIMSI – Groupe LIR
BP 133, 91403 Orsay
[nom]@limsi.fr

Résumé – Abstract

Nous proposons dans cet article un protocole permettant l'évaluation des analyseurs syntaxiques du Français. Le formalisme d'annotation retenu ici permet l'annotation à la fois des constituants et des relations fonctionnelles. Le corpus recueilli contient un assortiment de différents types de texte. Un sous-ensemble de ce corpus a été manuellement annoté. La précision, le rappel, le nombre de frontières croisées seront calculés pour 2 analyseurs qui participent actuellement au test du protocole l'un venant de l'industrie l'autre d'un organisme de recherche.

This paper presents PEAS, the first comparative evaluation framework for syntactic parsers of French whose annotation formalism allows the annotation of both constituents and functional relations. A test corpus containing an assortment of different text types has been built and part of it has been manually annotated. Precision/Recall and crossing brackets metrics will be adapted to our formalism and applied to the parses produced by one parser from academia and another one from industry in order to validate the framework.

Keywords – Mots Clés

Évaluation des analyseurs
Parser evaluation

1 Introduction

En traitement automatique du langage naturel, beaucoup d'applications complexes utilisent parmi leurs fonctionnalités de base un analyseur syntaxique. Il existe aujourd'hui pour le français un grand nombre d'analyseurs syntaxiques, certains se limitent au calcul des constituants, d'autres s'attachent en plus à déterminer les relations fonctionnelles entre ces constituants, abordant ainsi des problèmes liés à la sémantique. Devant cette diversité, il est

temps de proposer un cadre pour l'évaluation comparative de ces analyseurs ; ce cadre devant pour le moins contenir : un formalisme d'annotation, un vaste corpus partiellement annoté, les métriques permettant l'évaluation proprement dite ainsi que les outils associés.

Il est intéressant de souligner que la plupart des analyseurs récemment développés utilisent une approche robuste : ils ne construisent pas toujours une analyse complète de la phrase traitée, mais ils sont capables de construire un résultat quelles que soient la taille, les particularités, le niveau de grammaticalité du corpus à traiter. Une approche robuste étant particulièrement bien adaptée si l'on veut traiter de grandes quantités de données telles que celles manipulées sur le Web ou en recherche documentaire. Ces analyseurs robustes sont en outre susceptibles d'être des analyseurs partiels : ils ne couvrent pas tous les phénomènes syntaxiques. Notre protocole ne devant pénaliser aucune approche, il devra envisager une couverture syntaxique la plus complète possible : qui tienne compte du plus grand nombre possible de phénomènes traités par les analyseurs candidats, sans en pénaliser aucun.

Le but de ce travail est de proposer un tel cadre d'évaluation, car il n'en existe pas à ce jour pour le français. La figure 1 présente les différents modules de PEAS, notre protocole d'évaluation.

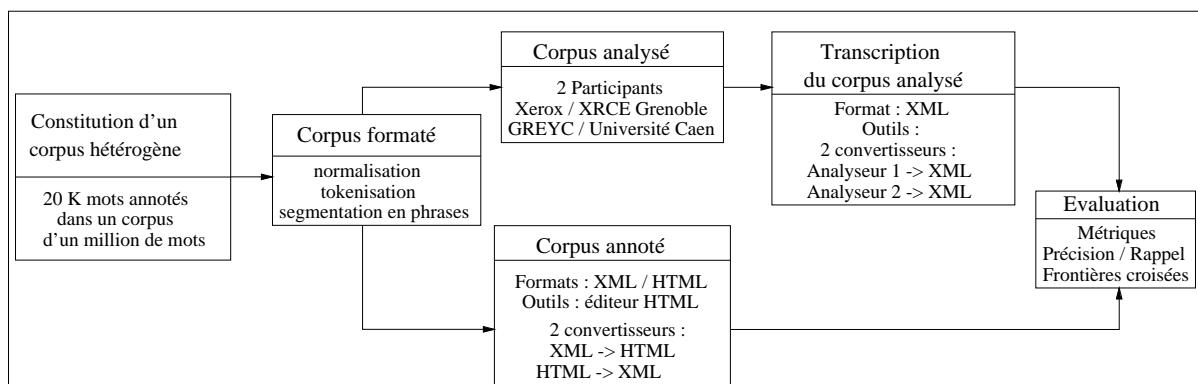


Figure 1: Modules du protocole d'évaluation

2 Formalisme d'annotation

D'un point de vue théorique, la définition du formalisme d'annotation est la tâche la plus complexe du protocole d'évaluation. En effet, le formalisme retenu doit permettre une couverture aussi large que possible des phénomènes syntaxiques afin de permettre à tout analyseur de participer dans de bonnes conditions à l'évaluation et ce quel que soit le formalisme grammatical qu'il utilise.

Très rapidement, nous nous sommes décidés pour une annotation en deux phases : tout d'abord les constituants (ou segments ou encore *chunks*) sont annotés et ensuite les relations fonctionnelles. Les relations peuvent associer des constituants, des mots, ou encore des mots et des constituants.

Les types de constituants annotés sont au nombre de six. Les constituants sont continus et ne sont pas imbriqués. Ils sont aussi petits que possible, afin de permettre et de faciliter la conversion des sorties des analyseurs dans notre formalisme. De plus, l'information qui ne serait pas exprimée dans les constituants ainsi découpés, est exprimée à travers un nombre important de relations fonctionnelles : onze au total. Un tel formalisme est plus proche d'un formalisme fondé sur les dépendances que d'un formalisme fondé sur les constituants (Sleator et Temperley, 1991). Il a pour intérêt de n'empêcher aucun analyseur même *profond* d'être évalué, cependant pour ce type d'analyseur la transcription des sorties sera plus complexe.

Les six types de constituants et les onze relations fonctionnelles sont donnés dans la table 1. Leur choix a été principalement inspiré des travaux d'Anne Abeillé et al. (2000) et d'une étude de corpus.

Constituants	Relations fonctionnelles
1) NV – noyau verbal	1) sujet-verbe
2) GN – groupe nominal	2) auxiliaire-verbe
3) GR – groupe adverbial	3) argument-verbe
4) GA – groupe adjectival	4) modifieur-verbe
5) GP – groupe prépositionnel introduisant un groupe nominal	5) modifieur-nom 6) modifieur-adjectif
6) PV – groupe prépositionnel introduisant un groupe verbal	7) modifieur-adverbe 8) attribut-sujet/objet 9) coordination
	10) apposition
	11) complémenteur

Table 1: Constituants et relations fonctionnelles

Notons que les propositions ne sont pas identifiées dans l'annotation des constituants. Mais, comme dans un formalisme à base de dépendances, la structure complexe de la phrase est obtenue à travers l'ensemble des relations fonctionnelles. Dans l'exemple suivant (Leroux 1907), on compte trois propositions (Pr) coordonnées : <Pr1> la porte de la chambre fermée à clef à l'intérieur </Pr1><Pr2> les volets de l'unique fenêtre fermés, eux aussi, à l'intérieur </Pr2> et <Pr3> par-dessus les volets, les barreaux intacts </Pr3>, [...]. Dans notre formalisme, ces trois propositions ne sont pas annotées en tant que telles, mais les relations qu'elles expriment sont bien représentées :

```
<GN1> la porte </GN1>
<GN2> les volets </GN2>
<GN3> les barreaux </GN3>
coordination (“,”, GN1, GN2)
coordination (et, GN2, GN3)
```

Et la première proposition Pr1 est exprimée par :

```
<GN1> la porte </GN1>
<GP1> de la chambre </GP1>
<GA1> fermée </GA1>
<GP2> à clef </GP2>
<GP3> à l'intérieur </GP3>
modifieur-nom (GP1, porte)
modifieur-nom (GA1, porte)
modifieur-adjectif (GP2, fermée)
```

De plus comme nos constituants ne sont pas imbriqués, tous les modificateurs placés devant un nom sont inclus dans le même groupe nominal que le nom lui-même. Et, là encore, les relations fonctionnelles sont utilisées pour exprimer les relations entre les termes, comme dans l'exemple qui suit, de l'annotation de *Mon très riche et très proche ami*.

```
<GN> mon très riche et très proche ami </GN>
modifieur-adjectif (très, riche)
modifieur-adjectif (très, proche)
coordination (et, riche, proche)
modifieur-nom (et, ami)
```

Le formalisme donne aussi la possibilité d'annoter des ambiguïtés au niveau des relations fonctionnelles (en dupliquant les tables décrivant ces relations) ; même si nous sommes encore en train d'étudier comment nous traiterons cela dans notre évaluation.

3 Corpus et outils d'annotation

Le corpus retenu pour l'annotation, est un ensemble de textes de natures aussi diverses que possible. En effet, celui-ci contient des extraits d'articles de journaux, de romans, de pages Web, de transcription de données audio et un ensemble de questions traduites des dernières campagnes TREC sur les systèmes de question-réponse. L'ensemble du corpus réunit un million de mots, chaque texte a été segmenté en phrases et en mots. Chaque participant a reçu les textes à analyser dans leur format original et dans cette forme segmentée.

La partie du corpus annotée contient environ 20 000 mots. Les outils d'annotation que nous avons développés utilisent un l'éditeur HTML Netscape Composer. Chaque constituant est sélectionné par l'annotateur puis coloré en fonction de son type (le passage en italique, marque la frontière entre deux constituants distincts de même type). Pour les onze relations fonctionnelles, l'annotateur dispose de onze tables qu'il remplit en indiquant soit le numéro du mot soit celui du constituant concerné. Toutes les tables ne sont heureusement pas à remplir

pour chaque phrase. Les informations ainsi annotées sont ensuite traduites dans un format XML, dont nous donnons ci-dessous un aperçu avec l'exemple repris du § 2 :

```

<E id="0">
<constituants>
<Groupe type="GN" id="G0">
<F id="F0"> la </F>
<F id="F1"> porte </F>
</Groupe>
<Groupe type="GP" id="G1">
<F id="F2"> de </F>
<F id="F3"> la </F>
<F id="F4"> chambre </F>
</Groupe>
<Groupe type="GA" id="G2">
<F id="F5"> fermée </F>
</Groupe>
<Groupe type="GP" id="G3">
<F id="F6"> à </F>
<F id="F7"> l' </F>
<F id="F8"> intérieur </F>
</Groupe>
<F id="F9"> , </F>
<Groupe type="GN" id="G4">
<F id="F10"> les </F>
<F id="F11"> volets </F>
</Groupe>
<Groupe type="GP" id="G5">
<F id="F12"> de </F>
<F id="F13"> l' </F>
<F id="F14"> unique </F>
<F id="F15"> fenêtre </F>
</Groupe> ...
</constituants>
<relations>
<rel xmlns:xlink="extended" type="MOD-N" id="R0">
<modifieur xmlns:xlink="locator" href="G1">
<nom xmlns:xlink="locator" href="F1">
</rel>
<rel xmlns:xlink="extended" type="MOD-N" id="R1">
<modifieur xmlns:xlink="locator" href="G2">
<nom xmlns:xlink="locator" href="F1">
</rel>
<rel xmlns:xlink="extended" type="MOD-A" id="R2">
<modifieur xmlns:xlink="locator" href="G3">
<adjectif xmlns:xlink="locator" href="F5">
</rel> ...
<rel xmlns:xlink="extended" type="COORD" id="R9">
<coordonnant xmlns:xlink="locator" href="F9">
<coord-g xmlns:xlink="locator" href="F1">
<coord-d xmlns:xlink="locator" href="F11">
</rel> ...
</relations>
</E>
```

En ce qui concerne le français, (Abeillé et al. 2000) est la seule tentative de réalisation d'un corpus annoté en syntaxe. Dans ce cas, le corpus est homogène du point de vue du type de texte, car il ne contient que des articles extraits du journal *Le Monde* ; il couvre cependant des domaines variés qui vont du sport à la politique. L'approche est ambitieuse et intéressante : le corpus contient 1 million de mots, 17 000 lemmes distincts, et il est annoté à la fois en morpho-syntaxe et en relations grammaticales.

4 Métriques d'évaluation

On trouve dans Parseval (Black et al., 1991), les premières propositions en évaluation des analyseurs syntaxiques. En 1998, Carroll et al. (1998) proposent à leur tour un nouveau schéma d'évaluation. Depuis, deux approches ont été dessinées. La première, comme dans Parseval, est fondée sur les frontières des constituants, elle calcule le rappel et la précision entre constituants de la clé et de la référence, ainsi que le nombre de constituants pour lesquels les frontières sont croisées (à nouveau entre clé et référence). Cette approche, même si elle a été l'objet de vives critiques (Gaizauskas 1998, Lin 1998), est toujours utilisée. La seconde approche se fonde sur les relations fonctionnelles, là encore on calcule le rappel et la précision. En consultant les actes de *Beyond Parseval* (2002), on pressent que cette approche sera de plus en plus celle utilisée par les systèmes faisant de l'auto-évaluation.

Comme notre formalisme d'annotation produit les deux types d'informations – les constituants et les relations fonctionnelles – notre module d'évaluation peut effectuer les calculs correspondants aux deux approches. Néanmoins, il faut souligner que d'un point de vue technique, la transcription des sorties des analyseurs est plus systématique pour les relations que pour les constituants. En effet, dans notre formalisme, les relations associent des mots des constituants ou des mots et des constituants, mais il est toujours possible de faire correspondre la clé et la référence, car on sait toujours à quel constituant un mot appartient. D'autre part, en ce qui concerne la segmentation, les frontières des constituants peuvent varier de façon significative d'un analyseur à l'autre. Il nous faut donc prévoir, soit un nombre important de règles de transformation, soit des méthodes d'évaluation suffisamment flexibles.

5 Perspectives

Fondé sur ces travaux préliminaires, un projet nommé EASY/EVALDA, de plus grande envergure sur l'évaluation des analyseurs syntaxiques a été accepté par TECHNOLANGUE (programme proposé par 3 ministères : celui de l'industrie, de la culture et de la recherche). Une large communauté francophone s'est déclarée intéressée par ce projet qui regroupe : quatorze participants (appartenant à des universités ou à des entreprises privées) prêts à faire évaluer leur analyseur et cinq fournisseurs de corpus, intéressés par l'annotation en syntaxe et en relation fonctionnelles de larges corpus. Cette communauté de chercheurs va contribuer à remettre en cause ou à enrichir les différents aspects de notre proposition de protocole : aussi bien le formalisme d'annotation que les outils ou les métriques retenues.

De plus la participation d'un nombre important d'analyseurs permettra la production d'une ressource linguistique validée. En effet, nous pourrons produire la fusion automatique des

données annotées par les analyseurs, puis nous pourrons mais cette fois de façon manuelle corriger les analyses divergentes¹.

Enfin, le format XML dans lequel nous traduisons les informations issues de l'analyse (non seulement les constituants mais aussi les relations fonctionnelles) constitue un format d'échange ouvert. Disposer d'un tel format est primordial pour les applications qui utilisent parmi leurs outils un analyseur syntaxique. Dans les applications du type question-réponse par exemple, un analyseur est nécessaire pour analyser les questions et aussi pour analyser les données recueillies susceptibles de contenir la réponse. Utiliser un format d'échange permet non seulement de tester plusieurs analyseurs, mais aussi de choisir pour chaque tâche particulière l'analyseur le plus adapté.

6 Conclusion

Tous les modules de notre protocole sont à ce jour prêts et disponibles, exception faite des outils de mesure. Les deux analyseurs candidats à cette première évaluation ont reçu les données à traiter et nous ont transmis leurs résultats. La phase de transcription de leurs résultats dans notre formalisme est pour ainsi dire terminée. Elle a soulevé plus de difficultés que prévu, et a nécessité une bonne connaissance des analyseurs candidats.

Cette première expérience nous a permis de valider notre guide d'annotation, même si de nombreux points sont encore à préciser, de tester nos outils et, plus généralement, de vérifier la faisabilité de notre approche. Notre but est de proposer un protocole d'évaluation le plus large possible et permettant une évaluation ciblée des analyseurs.

Références

Abeillé A., Clément L. et Kinyon A. (2000). Building a treebank for French. Actes de *Second International Conference on Language Resources and Evaluation (LREC)*, (1):87-94, Mai 2000, Athènes, Grèce.

Beyond Parseval – Towards Improved Evaluation Measures for Parsing Systems. Atelier de Third International Conference LREC. Las Palmas, Espagne. John Carroll editor.

Black E., Abney S., Flickenger D., Gdaniec C., Grishman R., Harrison P., Hindle D., Ingria R., Jelinek F., Klavans J., Liberman M., Marcus M., Roukos S., Santorini B., et Strzalkowski and T. (1991), A procedure for quantitatively comparing the syntactic coverage of english grammars. In DARPA, editor, Actes de *the Fourth DARPA Speech and Natural Language Workshop*, pp 306-311, Pacific Grove, California, February. Morgan Kaufmann.

Gaizauskas R., Hepple M. et Huyck H. (1998), A scheme for comparative evaluation of diverse parsing systems. Actes de *First International Conference on Language Resources and Evaluation (LREC)*, (1):143-149, Mai 1998, Grenade, Espagne.

¹ Monceaux (2002) propose un algorithme de *combinaison* qui fusionne en une analyse les sorties de plusieurs analyseurs

Gendner V., Illouz G., Jardino M., Monceaux L., Paroubek P., Robba I., Vilnat A. (2002), A Protocol for Evaluating Analyzers of Syntax (PEAS). Actes de *Third International Conference on Language Resources and Evaluation (LREC)*, Mai 2002, Las Palmas, Espagne.

Leroux G. (1907), *Le mystère de la chambre jaune*. L'Illustration, Paris.

Lin D. (1998), Dependency based method for evaluating broad-coverage parsers. *Natural Language Engineering* 4 (2):97-114.

Monceaux L. (2002), *Adaptation du niveau d'analyse des interventions dans un dialogue. Application à un système de question-réponse*. Thèse de l'université Paris 11, Décembre 2002.

Sleator D. et Temperley D. (1991), *Parsing English with a Link Grammar*. Rapport de recherche CMU-CS-91-196, Carnegie Mellon U., School of Computer Science, 91 p.

Carroll J., Briscoe T., Sanfilippo A. (1998), Parser Evaluation: a Survey and a New Proposal. Actes de *First conference LREC*, (1):447-454, Mai 1998, Grenade, Espagne.

TagChunker : mécanisme de construction et évaluation

Gil Francopoulo

Tagmatica
101 avenue de Saint-Mandé, 75012 Paris
www.tagmatica.com

Résumé

Après un descriptif de l'algorithme d'analyse, l'article décrit comment est développée et maintenue la grammaire du système. Puis, il sera montré que loin d'être un processus ajouté a posteriori, l'évaluation est une tâche récurrente sinon quotidienne qui oriente de manière précise le planning d'évolution du chunker et de la grammaire associée.

Mots Clés

Analyse en constituants, évaluation

1 Introduction

Elaborer un système d'analyse syntaxique qui construise une structure complète est une tâche ardue si nous visons à la fois une large couverture et une bonne qualité. Mais pour un certain nombre d'applications, comme l'extraction de termes ou l'indexation de textes, l'analyse complète n'est pas nécessaire : une analyse partielle suffit, du moment que les ambiguïtés lexicales sont résolues et que les groupes nominaux et prépositionnels élémentaires (non-récuratif) sont formés.

L'analyseur que nous présentons réalise deux tâches : l'étiquetage morpho-syntaxique (i.e. « Part of speech tagging ») et l'analyse syntaxique en syntagmes élémentaires (i.e. « chunking »). Notons que nous ne traitons pas les opérations de délimitation de propositions (i.e. « clause bracketing »).

La notion de « chunk », dans le cadre d'un analyse partielle a été introduite dans les années 90 par Abney (1991) : « the typical chunk consists of a simple content word surrounded by a constellation of function words, matching a fixed template ». Elle a été reprise et complétée dans (Abney 1996).

Pour une présentation simple et efficace de toutes ces notions, vous pouvez consulter (Vergne 2000). Si vous cherchez un état de l'art à jour avec des comparaisons entre

analyseurs, veuillez consulter (Monceaux 2002) et (Gala 2001). Pour un panorama un peu plus large, voir (Abeillé 2000).

2 La chaîne d'analyse

La totalité des outils de traitement de la langue de Tagmatica est organisée en une boîte à outils nommée TagTools (cf. www.tagmatica.com).

L'entrée de la chaîne d'analyse est un document rédigé en français ou en anglais.

Le document va traverser les modules de traitement suivants :

Etape-1 : Le détecteur de format est appliqué afin de déterminer quel est le format parmi 17 valeurs possibles. Ce sera par exemple : HTML, RTF ou bien XML.

Etape-2 : En fonction du format, un segmenteur (i.e. ‘tokenizer’) en phrases et en mots est appelé (6 segmenteurs existants). On notera que les marques typodispositionnelles, quand elles sont présentes, sont exploitées au même titre que la ponctuation. Ainsi, pour un format richement balisé comme HTML, les balises de marque de début de liste sont exploitées. A contrario, quand le format est moins riche, comme le format texte, le segmenteur dispose d'un nombre limité d'indices et n'exploite que la ponctuation. Notons que les marques typodispositionnelles ne sont pas transmises aux traitements ultérieurs.

Etape-3 : Un détecteur de langue est appliqué (10 langues reconnues).

Etape-4 : En fonction de la langue, un analyseur morphologique français ou bien anglais est appelé. En sortie, chaque mot est considéré comme mot simple ou bien composé. Chaque mot produit une ou plusieurs analyses. Chaque analyse comporte la forme lemmatisée et l'étiquette morpho-syntaxico-sémantique. Il y a 188 étiquettes différentes¹.

Etape-5 : Si un mot est inconnu, un ratraper de fautes nommé TagCorrector est appliqué. A la sortie, tous les mots possèdent au moins une analyse lexicale : il y a toujours un résultat. Quand il y a plusieurs résultats, nous avons affaire à une ambiguïté lexicale.

Etape-6 : TagChunker est appliqué sur chaque phrase.

¹ A titre de comparaison, le jeu d'étiquettes Grace comportait 312 valeurs (www.limsi.fr/tlp/grace et (Paroubek 2000))

3 Entrées-sorties de TagChunker

Nous venons de voir que l'entrée du chunker est une phrase². Les appels sont indépendants les uns des autres. Ce n'est pas un système à mémoire. La sortie du chunker est une suite plate de syntagmes élémentaires. Les syntagmes ne sont pas rattachés entre eux. On ne distingue pas les actants des circonstants. Ils sont appelés ‘chunks’. Pour une phrase donnée, la sortie est constituée :

- a) de la liste des chunks,
- b) du type de phrase : interrogative, déclarative etc.

Chaque chunk est composé :

- a) de la liste des mots lexicalement désambiguïsés du chunk. Le système détermine donc les frontières de groupes.
- b) du nom du chunk. C'est une valeur à prendre parmi 21 étiquettes. Ce sera par exemple, GV, GN, GP ou GpotentiellementNominalOuPrépositionnel.

4 Algorithme d'analyse

Un automate de reconnaissance est appliqué sur la phrase³ :

Si un seul résultat est produit

Alors, il est considéré comme étant le bon résultat

Si plus d'un résultat est trouvé

Alors, un algorithme d'élagage est appliqué pour ne retenir qu'un seul résultat

Si il n'y a aucun résultat

Alors, des micro-grammaires sont combinées afin de produire des analyses candidates

 Si un seul résultat est produit

 Alors, il est considéré comme étant le bon résultat

 Si plus d'un résultat est produit

 Alors, l'élagage est appelé afin de ne retenir qu'un seul résultat

 Sinon, c'est un échec.

² Le reste de l'article ne traite que du français, mais le dispositif est très similaire pour l'anglais.

³ Nous verrons plus loin comment celui-ci est construit. Le présent chapitre se limite à son usage.

Un mot de la combinaison des micro-grammaires : nous faisons l'hypothèse qu'il existe une suite (idéale) de micro-grammaires capable d'analyser entièrement la phrase. Comme nous ne connaissons pas précisément cette suite idéale, nous procédons à toutes les permutations et nous élaguons.

L'algorithme actuel d'élagage est une cascade de filtres statistiques qui combinent les informations suivantes :

- a) les ambiguïtés entre étiquettes : étant donnée une combinaison d'étiquettes, quelle est l'étiquette la plus probable ?
- b) les bigrammes sur les suites de chunks. Quand on a un chunk de tel type, quelle est la probabilité pour qu'il soit suivi par un chunk de tel type ?
- c) les ambiguïtés lexicales des graphies : étant donnée une chaîne de caractères, quelle est la probabilité qu'elle ait telle analyse ?
- d) l'analyse qui propose le nombre minimal de chunks est favorisée. Ce qui revient à privilégier les chunks qui couvrent un nombre important de mots.

En résumé, on peut exprimer quatre choses importantes :

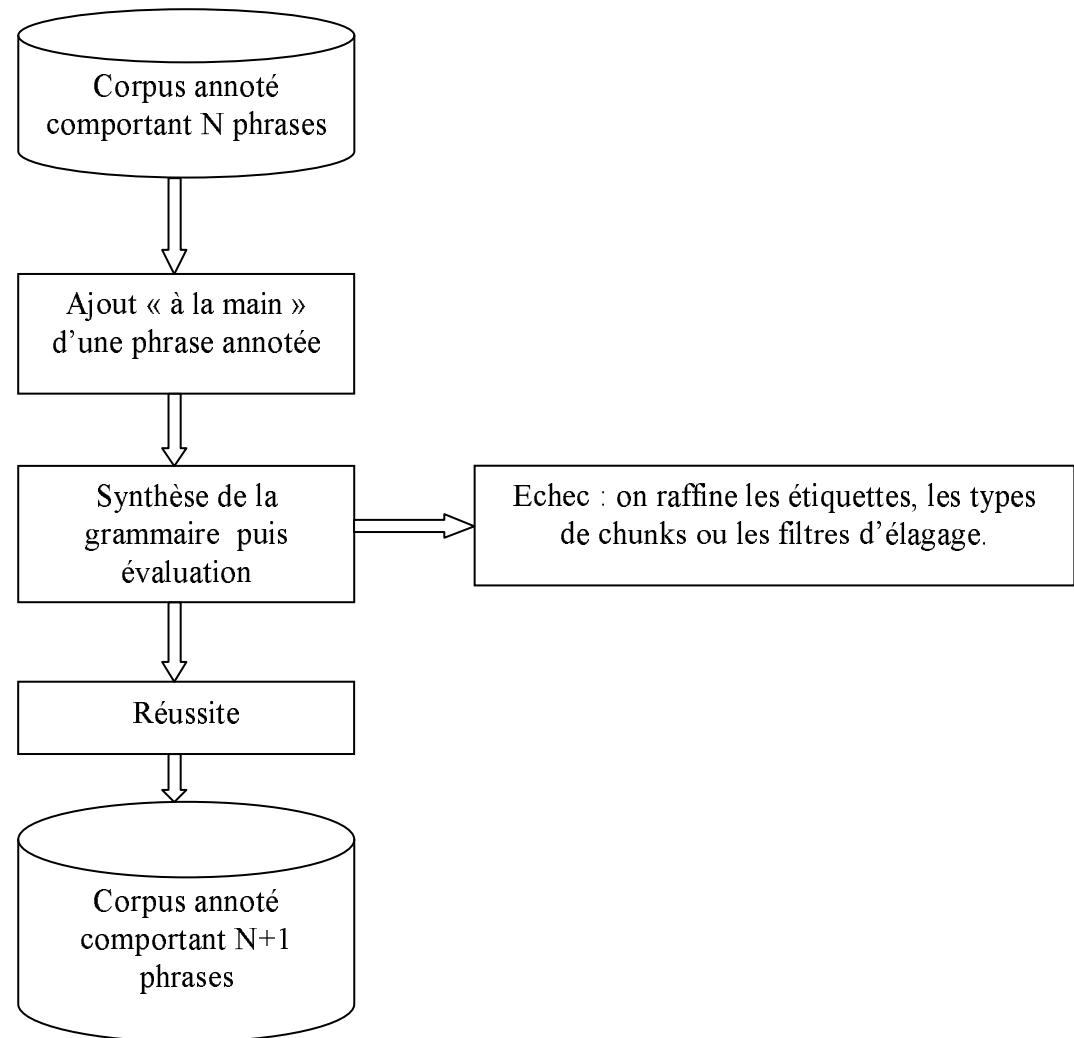
- a) l'algorithme d'élagage conditionne grandement la qualité des analyses.
- b) le chunker est un dispositif hybride mi-symbolique mi-statistique. On peut le décrire par la maxime suivante : « on essaie de manière symbolique, si cela ne fonctionne pas, on essaie de manière statistique ».
- c) il n'y a pas une phase de « tagging » puis en séquence, une phase de « chunking », comme dans certains systèmes. C'est le « chunking » qui pilote le choix des étiquettes et donc réalise le « tagging ».
- d) même si le sujet de l'article n'est pas de présenter en détail le jeu des 188 étiquettes, le choix de ces étiquettes est crucial pour l'obtention d'une bonne analyse. Certaines étiquettes sont affectées à beaucoup de mots, mais d'autres ne sont affectées qu'à un nombre très restreint de mots. Les étiquettes sont très précises pour les mots grammaticaux du français. Le rôle du lexique est donc loin d'être négligeable car les étiquettes agissent comme des déclencheurs sur des micro-grammaires (c.f. chapitre sur le corpus).

5 Mécanisme de construction de la grammaire

Par convention, nous appelons grammaire, le triplet formé par :

- a) l'automate de reconnaissance. C'est la représentation de l'ordre dans lequel chaque nom de chunk apparaît dans le corpus d'apprentissage.
- b) les micro-grammaires. Chacune d'elle reconnaît les chunks d'un certain type. Une micro-grammaire est composée du nom de chunk qu'elle reconnaît et de la liste des suites des étiquettes qui composent le chunk. Une suite d'étiquettes représente une analyse possible. Il y a 21 micro-grammaires.
- c) les matrices statistiques destinées à l'algorithme d'élagage.

La grammaire n'est pas directement écrite par un humain. Un corpus de phrases a été annoté avec des marques morpho-syntaxiques et c'est ce corpus qui est transformé en grammaire⁴. Le corpus annoté comporte 18 000 mots. Le corpus a été annoté entièrement à la main avec beaucoup d'attention, puis vérifié par un programme de la manière suivante :



L'automate est en fait la totalité des chemins d'analyse en conservant l'ordonnancement des types de chunks. L'évaluation consiste à appliquer la grammaire apprise sur le corpus initial : le même résultat doit être obtenu.

En ce qui concerne la méthodologie de construction, les étiquettes, les types de chunks et les filtres d'élagage ont été obtenus par essai-erreur.

⁴ Quelques idées ont été reprises de (Francopoulo 1986).

6 Choix et évolution du corpus d'apprentissage

Le corpus est envisagé comme moyen de décrire une grammaire. Lorsque le travail sur TagChunker a commencé, il n'existait pas de corpus français annoté en syntaxe comme il en existe en anglais (Marcus 93) (Sampson 95)⁵. Nous avons donc annoté nous-même un corpus. Même si un corpus avait existé, il aurait sans doute fallu raffiner l'annotation du fait de la finesse de notre jeu d'étiquettes. Le corpus d'apprentissage est composé de trois parties :

a) corpus noyau.

Ce sont des phrases inventées simples qui sont organisées par phénomène syntaxique. Il y a par exemple un fichier des groupes nominaux, un fichier des groupes adjectivaux, un fichier pour les dates etc. Ce corpus permet ainsi, par le biais des exemples, de poser une grammaire élémentaire du français en combinant des micro-grammaires.

b) corpus par échantillons pris au hasard.

Ce sont des phrases prises au hasard dans des textes législatifs, des romans, des journaux et des dépêches de presse. Les structures syntaxiques concernées sont relativement différentes les unes des autres. La longueur des phrases est très variable. Ce corpus a été ajouté dans une deuxième phase, donc après avoir établi une grammaire noyau, afin d'injecter des phrases complexes dans l'apprentissage. L'objectif est double, il s'agit d'une part de compléter les chunks « oubliés » dans la grammaire noyau, mais surtout de déclarer les ordonnancements possibles des phrases complexes.

c) corpus des échecs.

Une fois, les deux corpus précédents stabilisés, un troisième corpus a été ajouté pour traiter les cas non couverts par les corpus précédents. Ce corpus est constitué de dépêches de presse. En effet, l'annotation est très fastidieuse et continuer à annoter des phrases prises au hasard aurait été contre-productif. Il est en effet plus intéressant de traiter un cas non couvert plutôt que d'annoter une phrase dont la structure syntaxique est déjà décrite.

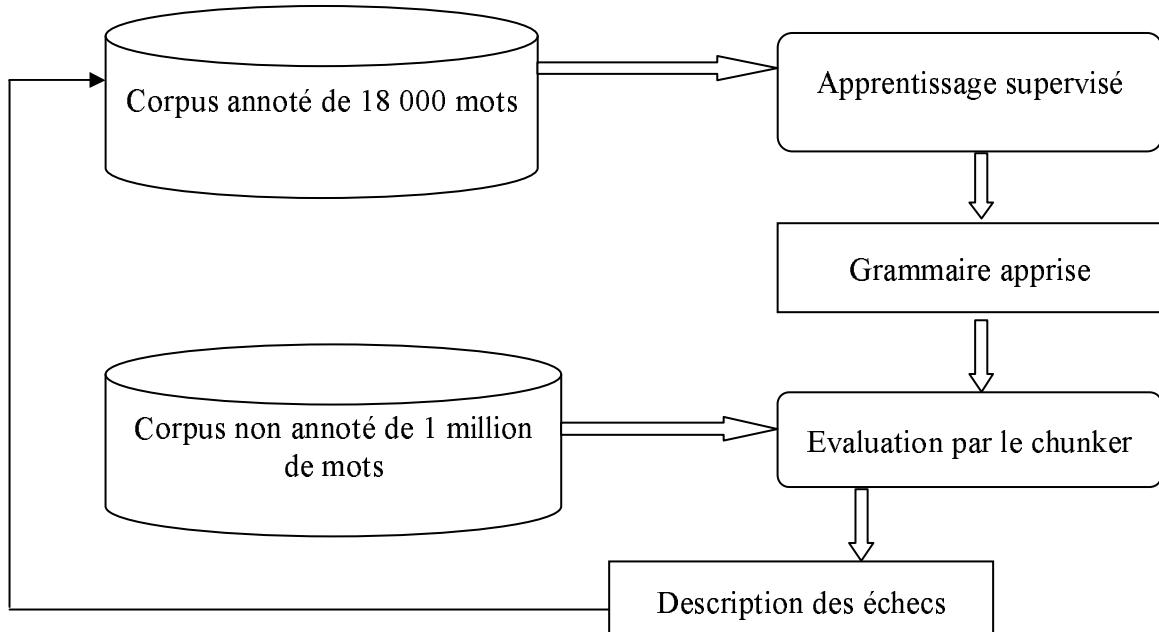
Voici le dénombrement du corpus :

	Nb de phrases	pourcentage
Corpus noyau	565	52 %
Corpus des échantillons	434	40 %
Corpus des échecs	86	8 %
total	1085	100 %

Actuellement, le seul corpus en augmentation est le corpus des échecs et cette opération vient de commencer. La démarche est itérative et repose sur une évaluation continue via le chunker. On appelle « évaluation », l'analyse de chaque phrase par la grammaire apprise. Si l'analyse est correcte, on se contente de la compter. Si l'analyse est incorrecte, on la compte et on mémorise la phrase dans un index.

⁵ Un corpus issu du journal « Le Monde » est en cours de préparation (Abeillé 2001). Si les conditions de son obtention sont acceptables, il pourra être substitué ou ajouté à notre corpus.

Les opérations peuvent être schématisées par le diagramme suivant :



Il n'est pas très facile de choisir parmi les échecs, la phrase qui apportera l'information la plus intéressante pour le système, car pour chaque échec, nous ne connaissons pas (automatiquement) la bonne analyse. En nous fondant sur le fait que plus la phrase est longue, plus elle est pénible à annoter, il vaut mieux sélectionner les phrases les plus courtes pour les annoter en priorité : c'est ce que nous faisons. Un autre critère nous a conforté dans notre choix est qu'en s'intéressant aux phrases courtes qui posaient problème, nous avons trouvé immédiatement les erreurs produites par la segmentation. Il y en avait une cinquantaine au départ.

7 Evaluation

Le rappel est défini comme le nombre d'analyses correctes produites sur le nombre d'analyses demandées. De manière plus intuitive, le rappel mesure le silence (en fait son inverse) dont fait preuve le système quand il ne trouve aucune analyse. La précision est définie comme le nombre d'analyses correctes produites sur le nombre d'analyses produites. La précision mesure donc le bruit produit par le système. Autrement dit, si l'on définit :

$$CP = \text{nombre d'analyses correctes produites}$$

$$NP = \text{nombre d'analyses non produites}$$

$$IP = \text{nombre d'analyses incorrectes produites}$$

Alors, on pose : $Rappel = CP/(CP+NP)$ et $\text{Précision} = CP/(CP+IP)$

Et, pour disposer d'une valeur globale (mais grossière) on calcule la F-Mesure comme étant :

$$FM = (2 * Rappel * \text{Précision}) / (Rappel + \text{Précision})$$

Nous avons trois types d'évaluation :

- a) l'évaluation lors de l'ajout d'une phrase. On s'assure que le système s'auto-applique avec succès à son propre corpus d'apprentissage.
- b) l'évaluation sur le grand corpus non annoté. La mesure fournie permet de connaître le rappel, par le biais du taux d'échecs, mais nous ne pouvons pas connaître la précision.
- c) l'évaluation sur un corpus annoté de tests. L'annotation est identique à l'annotation du corpus d'apprentissage. Les phrases ont été choisies au hasard dans un roman et des dépêches d'agence. Sa taille est d'environ 10% du corpus d'apprentissage. Cette évaluation permet de mesurer à la fois le rappel et la précision. C'est d'ailleurs la seule mesure de précision que nous ayons.

	Nb de mots	Rappel	Précision	F-Mesure
Corpus d'apprentissage	17,0 K	100 %	100 %	100 %
Corpus non annoté	1025,0 K	80 %	non calculée	non calculée
Corpus de test	1,5 K	83 %	66 %	74 %

On observe que la précision n'est pas très bonne, mais la tâche est difficile. Les phrases en question ne sont pas des phrases jouets. Ce sont des phrases réelles. Certaines de ces phrases sont complexes et le test est relativement sévère. Une phrase est considérée comme bonne si elle remplit les trois conditions suivantes :

- a) la désambiguïsation lexicale (i.e. tagging) doit être correcte.
- b) les frontières de mots doivent être correctement déterminées.
- c) les noms des chunks doivent être correctement fixés.

S'il s'agissait de mesurer la seule désambiguïsation lexicale comme dans la campagne d'évaluation Grace (Paroubek 2000), les résultats seraient évidemment meilleurs. Faute de temps, nous ne les avons pas calculés.

8 Utilité des évaluations

L'objectif n'est pas de comparer notre système à un autre analyseur comme dans la campagne Evalda-Easy. Nous réalisons des évaluations de manière continue afin de faire émerger les situations suivantes :

- a) détecter les circonstances de régression. Il ne faut pas que l'ajout d'annotations ou des modifications dans les filtres d'élagage dégradent la qualité du résultat.
- b) permettre de décider quelles sont les actions à privilégier. En général, il faut choisir entre le rappel, la précision et la vitesse de traitement.

Notre démarche ne procède pas en deux étapes : i) l'élaboration du système, ii) son évaluation. Au contraire, l'évaluation est quotidienne. Elle vérifie et dirige l'élaboration.

9 L'avenir

L'évaluation décrite ici est le mécanisme actuellement mis en oeuvre. L'observation immédiate que l'on peut tirer du tableau est que l'on ne mesure pas très bien la précision du fait de la taille réduite du corpus de test⁶. En revanche, la mesure de rappel est beaucoup plus fiable. Si dans quelques mois, dans le cadre de la campagne d'évaluation Evalda-Easy, il est possible d'avoir accès à un corpus annoté de bonne qualité, nous pourrons alors mesurer plus sereinement la précision.

A court terme, le corpus annoté va être augmenté des groupes non encore reconnus. Il s'agira d'améliorer le rappel, et ainsi élargir la couverture. Au lieu de se focaliser sur les échecs, il semble possible d'évaluer groupe par groupe pour détecter sur quel type de phénomène syntaxique, le système doit être amélioré. Le problème est que le temps est compté : plus on passe de temps à raffiner l'évaluation, moins il en reste pour annoter le corpus. A long terme, il n'est pas très clair s'il est plus intéressant d'améliorer le rappel et la précision en conservant le mécanisme actuel, ou bien s'il faut se préoccuper de produire une analyse un peu plus complète comme un regroupement des chunks en clauses. C'est ce que font les systèmes IPS (Wehrli 92), IFSP (Aït-Mokthar 97), XIP ou l'analyseur du GREYC (Giguet 97).

Suivant Abney, la sélection lexicale semble la technique la plus prometteuse : « The relationships between chunks are mediated more by lexical selection than by rigid templates » (Abney 91). Il s'agirait alors de combiner des structures actancielles (Tesnière 59) lexicalisées avec une grammaire des subordonnées et des circonstants. Parce que, même si l'usage externe du système est de « tagger » des textes ou bien de réaliser un « chunking », il n'est pas interdit au système de faire appel, de manière interne à des connaissances qui dépassent ces niveaux d'analyse. On peut imaginer un système d'analyse syntaxique et sémantique robuste dont le seul résultat exploité est une désambiguisation lexicale, seulement un tel système aurait une fenêtre d'analyse et donc une qualité bien supérieure à un tagger fondé sur des trigrammes lexicaux par exemple. Mais il faut éviter, que sous prétexte de réaliser une analyse en profondeur, le système devienne fragile. L'analyseur idéal est un système qui analyse en profondeur quand c'est possible, et dans le cas contraire, se comporte comme un chunker.

10 Conclusion

L'évaluation est utile car nous constatons que les actions menées jusqu'à présent ont amélioré le système. Et nous posons l'hypothèse que le système peut encore être amélioré grâce à ce genre d'action. On s'arrête quand il n'y a plus de progression : on est alors dans un optimum local ou global. Mais, pour un changement radical (l'ajout d'un niveau d'analyse, par exemple) l'argument ne tient plus : la chose mesurable n'a pas de passé puisqu'elle n'existe pas du tout.

En conclusion, pour une décision à long terme, au contraire d'une amélioration locale, l'évaluation n'est pas d'un grand secours.

⁶ Il serait possible de tirer au hasard des phrases du corpus d'apprentissage pour les injecter dans le corpus de test, mais cela affaiblirait d'autant la couverture de la grammaire.

Références

- Abeillé A., Blache P. (2000) Grammaires et analyseurs syntaxiques. *Ingénierie des langues*. Pierrel, J.M., (eds) Hermès.
- Abeillé A., Clément L., Kinyon A., Toussenel F. (2001) Un corpus français arboré : quelques interrogations. Actes de *la conférence sur le traitement automatique de la langue naturelle*, TALN, Tours.
- Abney S. (1991) Parsing by chunks. In Berwick, R., Abney, S., Tenny, C. (eds) *Principle-based parsing*. Kluwer.
- Abney S. (1996) Part-of-speech tagging and partial parsing. In Church, K., Young, Y., Blothoof, G., (eds) *Corpus-based methods in Language and speech*. Kluwer.
- Aït-Mokthar S., Chanod J. (1997) Incremental finite-state parsing. In *Proceeding of ANLP-97*, Washington.
- Francopoulo G. (1986) Machine learning as a tool for building a deterministic parser. In *Proceeding of GWAI-86*. Springer-Verlag.
- Gala Pavia, N., (2001) A two-tier corpus-based approach to robust syntactic annotation of unrestricted corpora. In Daille, B., Romary, L. (eds) *Linguistique de corpus, Traitement automatique des langues Vol 42*. Hermès.
- Giguet E., Vergne J. (1997) From part-of-speech tagging to memory-based deep syntactic analysis. In *Proceeding of IWPT'97*, Boston, Massachussets.
- Marcus M.P., Santorini B., Marcinkiewicz M.A. (1993) Building a large annotated corpus of english : the Penn treebank. *Computational Linguistics*, 19.
- Monceaux L. (2002) Adaptation du niveau d'analyse des interventions dans un dialogue : application à un système de question-réponse. Thèse de l'Université Paris 11, Décembre 2002.
- Paroubek P., Rajman M. (2000) Etiquetage morpho-syntaxique. *Ingénierie des langues*. Pierrel, J.M. (eds) Hermès.
- Sampson G. (1995) English for the computer : The susanne corpus and analytic scheme. Clarendon Press Oxford.
- Tesnière L. (1959) Eléments de syntaxe structurale. Kiencksieck.
- Vergne J. (2000) Trends in robust parsing. Actes de *Tutorial-Coling 2000*.
- Wehrli E. (1992) The IPS system. Actes de *Proceeding of Coling 1992*.

TALN et multilinguisme
organisée par Malek Boualem et Emilie Guimier-De-Neef
(France Télécom R&D - DMI/GRI)

Multilinguisme et question-réponse: adaptation d'un système monolingue

Luc Plamondon (1), George Foster (2)

(1) RALI - Université de Montréal
Montréal, Québec, Canada
plamondl@iro.umontreal.ca

(2) RALI - Université de Montréal
Montréal, Québec, Canada
foster@iro.umontreal.ca

Mots-clefs – Keywords

Question-réponse multilingue, extraction d'information, recherche d'information translinguis-tique, traduction automatique

Multilingual Question Answering, Information Extraction, CLIR, Machine Translation

Résumé - Abstract

Nous montrons comment il est possible de modifier un système de question-réponse monolingue anglais afin de répondre à des questions posées en français. Nous avons écrit de nouvelles règles afin d'analyser la question en français et nous avons utilisé un moteur de traduction proba-biliste pour traduire les termes-clés. Si l'implantation du bilinguisme a entraîné une diminution de performance, cette approche donne de meilleurs résultats qu'un système monolingue couplé à un système de traduction automatique pour traduire les questions.

We describe a method for modifying a monolingual English question-answering system in order to accept French questions. Our method relies on a statistical translation engine to translate keywords, and a set of manually-written rules for analyzing French questions. Although adding bilingualism causes a drop in performance, our method yields better results than a baseline approach of translating French questions automatically before submitting them to the original monolingual system.

1 Introduction

Un système de question-réponse (QR) est un type particulier de moteur de recherche qui permet à un utilisateur de poser une question en langue naturelle, plutôt que de l'obliger à construire une requête en un langage artificiel fait de ET, OU et autres opérateurs. De plus, alors qu'un moteur de recherche classique propose à l'utilisateur une liste des documents les plus pertinents à sa requête, le système de QR extrait lui-même les réponses aux questions. La recherche s'effectue dans un ensemble de documents ou de pages web.

Clarke et ses collaborateurs ont montré que pour des collections de textes totalisant moins de 500 gigaoctets (100 milliards de mots), plus la collection est volumineuse, meilleures sont les performances de leur système de QR (Clarke *et al.*, 2002). Si l'on suppose que les anglophones ont accès à environ 10 fois¹ plus de documents numérisés — pages web, encyclopédies sur cédéroms, etc. — que les francophones, il ne fait aucun doute qu'un système de QR destiné aux francophones mais capable de fouiller des textes en anglais ouvre des perspectives intéressantes à la fois au niveau de la quantité des sujets couverts et de la qualité des réponses trouvées.

Dans le cadre des campagnes d'évaluation TREC, nous avons développé le système de QR Quantum (Plamondon *et al.*, 2002). Ce système fonctionne uniquement en anglais: la question doit être posée en anglais, les documents de la collection sont entièrement en anglais et l'extraction des réponses nécessite des ressources linguistiques en anglais. Nous avons récemment entrepris de rendre Quantum bilingue, de sorte qu'un utilisateur francophone puisse poser sa question en français et obtenir une réponse en français extraite d'une collection de textes en anglais. Nous mettrons en évidence les problèmes que pose le multilinguisme en QR et nous décrirons comment nous avons choisi de les résoudre dans Quantum.

2 Travaux antérieurs

La campagne d'évaluation TREC-8 (*Text Retrieval Conference*, 1999) et les suivantes ont favorisé la création de quantité de systèmes de QR pour l'anglais. Bien que les techniques diffèrent d'un système à l'autre, la plupart suivent ces trois étapes: analyse de la question pour déterminer le type de réponse attendu, filtrage de la collection de textes pour ne conserver que les plus pertinents et identification de la réponse exacte.

Il existe aussi des systèmes monolingues pour l'italien (Magnini *et al.*, 2001), le polonais (Vetulani, 2002), le japonais (Seki & Harada, 2002) et le coréen (Kim & Seo, 2002). Quant aux systèmes pour le français, la campagne d'évaluation EQuER en cours de préparation, dans le cadre du projet EVALDA de Technolangue/ELDA, donnera sans doute lieu à la même récolte que les campagnes TREC. Il n'existe pas encore, à notre connaissance, de système multilingue qui permette de poser une question dans une langue différente de la collection de textes et d'obtenir la réponse dans la même langue que la question. Le nouveau volet de la campagne d'évaluation annuelle CLEF destiné aux systèmes de QR multilingues favorisera sûrement la conception de systèmes de ce genre (la traduction de la réponse n'étant toutefois pas à l'ordre du jour).

¹Estimation basée sur la proportion de pages web en chacune de ces langues: selon la firme Global Reach, spécialisée en marketing en ligne international, 40,2% de toutes les pages web sont en anglais contre 3,9% en français (www.global-reach.biz/globstats).

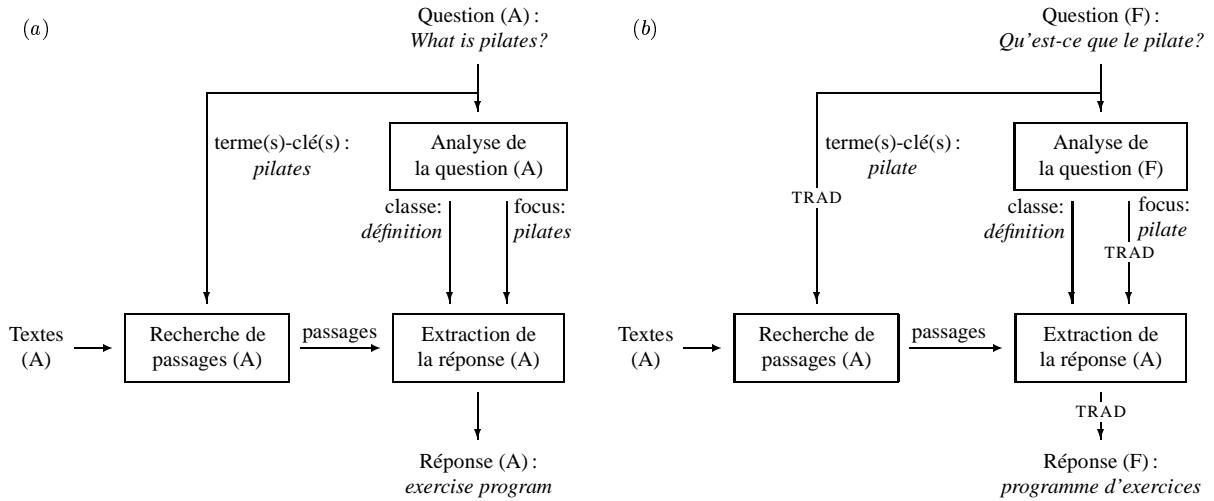


Figure 1: Architecture des deux versions de Quantum. (a) Version monolingue: la question, les textes et la réponse sont en anglais (b) Version bilingue: la question et la réponse sont en français (F). Le module d'analyse des questions a été francisé et les autres modules opèrent en anglais (A). La traduction se fait en 3 points: termes-clés, focus et réponse.

3 Système monolingue

Quantum a d'abord été développé en fonction des campagnes TREC. Il a été conçu pour répondre à des questions en anglais, simples, courtes, factuelles (c'est-à-dire que les réponses sont la plupart du temps des entités nommées) et syntaxiquement bien formées comme *What is pilates?* *Who was the architect of Central Park?* *How wide is the Atlantic Ocean?* *At what speed does the Earth revolve around the sun?* *Where is the French consulate in New York?* La collection de textes est constituée d'environ un million de dépêches de différentes agences de presse de langue anglaise, pour un total de 3 gigaoctets de données (600 millions de mots). Les réponses sont extraites des documents et doivent être *exactes*, c'est-à-dire que la chaîne de caractères suggérée en guise de réponse ne doit contenir rien de plus ou de moins que la réponse. Pour de plus amples détails, le lecteur est invité à consulter la description de la piste *Question Answering* de TREC-11 (Voorhees, 2002).

L'architecture du système monolingue est exposée à la figure 1a. Nous ne décrirons ici que les éléments pertinents à l'implantation du bilinguisme (figure 1b).

3.1 Analyse de la question

Le but de l'analyse de la question est de déterminer le type de la réponse à chercher, donc le mécanisme d'extraction — ou fonction d'extraction — à utiliser. Parfois, il est nécessaire de transmettre à cette fonction un paramètre appelé *focus* de la question: il s'agit d'un mot ou groupe de mots présents dans la question et ayant un rapport étroit avec la réponse. Par exemple, la réponse à la question *What card game uses only 48 cards?* doit être un hyponyme du focus *card game*, c'est pourquoi il est nécessaire de transmettre ce paramètre à la fonction d'extraction

choisie. Quant à la réponse à la question *How many black keys are on the piano?*, elle doit contenir un nombre suivi d'une répétition du focus *black keys*. Cependant, certaines autres questions comme *When was water found on Mars?* ne nécessitent pas l'identification d'un focus puisqu'il s'agit ici de trouver une entité nommée de temps pour compléter le concept désigné par *when*. Peu importe qu'ils soient focus ou non, tous les mots de la question contribuent à l'identification de la réponse par le biais du score de la recherche d'information (section 3.2). Il est important de souligner que dans la classification des questions que nous avons retenue, la nécessité ou non de passer un focus à telle ou telle fonction d'extraction ainsi que les critères de sélection du focus sont motivés par des considérations purement techniques et propres au fonctionnement interne de Quantum. Une classification plus générale basée sur des critères psycho-linguistiques rigoureux a été faite par Graesser (*Graesser et al.*, 1992).

L'analyse de la question s'effectue à l'aide d'un segmenteur de mots, d'un étiqueteur grammatical statistique ainsi que d'un segmenteur de groupes nominaux basé sur les étiquettes grammaticales. Nous avons élaboré une soixantaine de règles d'analyse utilisant à la fois des mots, des étiquettes grammaticales et des étiquettes de groupes nominaux (Plamondon, 2002). Par exemple, la règle d'analyse qui s'applique à la question de la figure 1a est la suivante:

```
what BE <syntagme nominal SN1> → type = définition, focus = SN1
```

3.2 Filtrage des passages les plus pertinents

Les mécanismes d'extraction de réponse sont trop complexes pour être appliqués à la totalité de la collection de textes à chaque question. Afin de réduire l'espace de recherche de 3 milliards de caractères à quelques milliers, nous utilisons un moteur de recherche conventionnel qui ordonne les passages les plus pertinents. Nous utilisons à cet effet le moteur Okapi car il peut raffiner sa recherche jusqu'à suggérer les paragraphes les plus pertinents, à la différence des autres moteurs de recherche qui s'en tiennent à une liste de documents complets. Il nous suffit ensuite de conserver les 20 meilleurs paragraphes et le score de pertinence qu'Okapi leur attribue. La requête est formée de tous les mots de la question; Okapi se charge de les tronquer et d'éliminer les mots vides (*stopwords*) avant d'effectuer la recherche. La requête de l'exemple de la figure 1a est composée du seul mot de la question qui ne soit pas vide de sens: *pilates*.

3.3 Extraction de la réponse

La fonction d'extraction choisie lors de l'analyse de la question, paramétrée ou non avec le focus, est appliquée aux paragraphes les plus pertinents (Plamondon, 2002). Trois techniques ou outils peuvent être utilisés par l'une ou l'autre des 12 fonctions d'extraction: des expressions régulières, le réseau sémantique Wordnet et l'extracteur d'entités nommées Annie de la suite de développement GATE. Par exemple, il est souhaitable que la réponse à une demande de définition contienne un hypéronyme du focus; ainsi, pour répondre à la question *What is ouzo?*, Wordnet nous permet de vérifier que la réponse *liquor* est bien un hypéronyme du focus *ouzo*. Cependant, dans l'exemple à la figure 1a, la réponse doit être trouvée à l'aide d'expressions régulières car Wordnet n'a pas d'entrée *pilates*, le focus de la question.

Chaque groupe nominal dans les paragraphes retenus se voit attribuer un score d'extraction selon qu'il satisfait aux critères de la fonction d'extraction. Ce score d'extraction est combiné

au score de pertinence du paragraphe afin de tenir compte de la densité des mots de la question se trouvant autour du groupe nominal examiné. Le meilleur groupe nominal est sélectionné pour constituer la réponse. Nous avons choisi de considérer les groupes nominaux comme unités de base de réponse car il s'avère que seulement 2 % des questions des campagnes TREC ne peuvent être répondues à l'aide d'un groupe nominal.

4 Implantation du bilinguisme

Pour Quantum comme pour bien d'autres systèmes de QR, l'étape d'extraction de la réponse est l'étape la plus complexe. Elle est donc déterminante dans le choix de l'approche à favoriser lorsqu'il s'agit d'adapter un système monolingue à des questions ou à une collection de textes d'une autre langue. Deux facteurs sont à considérer plus particulièrement: la disponibilité de ressources linguistiques nécessaires à l'extraction de la réponse et la facilité d'implantation du multilinguisme.

Dans le cas de Quantum, ces deux facteurs ont joué en faveur de la conservation du module d'extraction en anglais (figure 2a) et, par conséquent, en faveur de la traduction de la question et de la réponse plutôt que des textes. D'une part, les ressources linguistiques sont généralement plus nombreuses et plus poussées en anglais qu'en français, si ce n'est qu'elles sont disponibles gratuitement (c'est le cas de Wordnet et de l'extracteur d'entités nommées Annie auxquels Quantum fait appel). D'autre part, en laissant le module d'extraction dans sa langue d'origine, l'implantation du bilinguisme s'en trouve facilitée. En effet, si nous avions plutôt choisi de transposer le module d'extraction dans la même langue que celle de la question (figure 2b), il aurait fallu trouver de nouvelles ressources linguistiques, adapter le système à leur interface et traduire des textes entiers de l'anglais au français, ce qui pour le moment demeure une entreprise longue et ardue. Par contre, ne traduire que la question et la réponse est plus facile; nous verrons plus loin qu'il n'est pas nécessaire de parvenir à une traduction syntaxiquement correcte de la question et que la traduction de la réponse peut tirer parti du contexte particulier de la QR.

Cependant, dans le cas où la langue d'origine du système s'avère être la plus pauvre en ressources linguistiques, comme ce pourrait être le cas si l'on voulait rendre bilingue un système monolingue français, les deux facteurs à considérer jouent l'un contre l'autre. Il faudrait choisir entre une implantation rapide en laissant le cœur du système en français (figure 2b) et un gain potentiel de performance en transformant son cœur pour qu'il utilise des ressources linguistiques en anglais (figure 2a).

Si l'enjeu était plutôt d'adapter un système monolingue anglais non pas à des questions posées dans une langue différente mais bien à des textes écrits dans une langue différente, des considérations supplémentaires devraient être prises en compte. En effet, laisser le cœur dans sa langue d'origine (figure 3a) faciliterait l'implantation du bilinguisme mais obligerait à traduire de longs textes. Cependant, en utilisant des moteurs de traduction probabilistes, il serait relativement simple d'adapter le système à une collection regroupant des textes en plusieurs langues. L'approche 3b, quant à elle, dépendrait moins du moteur de traduction mais nécessiterait un système différent pour chaque langue rencontrée dans la collection.

Dans les scénarios présentés, le cœur du système est considéré comme une boîte noire hermétique au processus de traduction; ceci suppose une traduction parfaite des questions ou des textes. Les méthodes de traduction actuelles ne le permettent évidemment pas. Il est donc par-

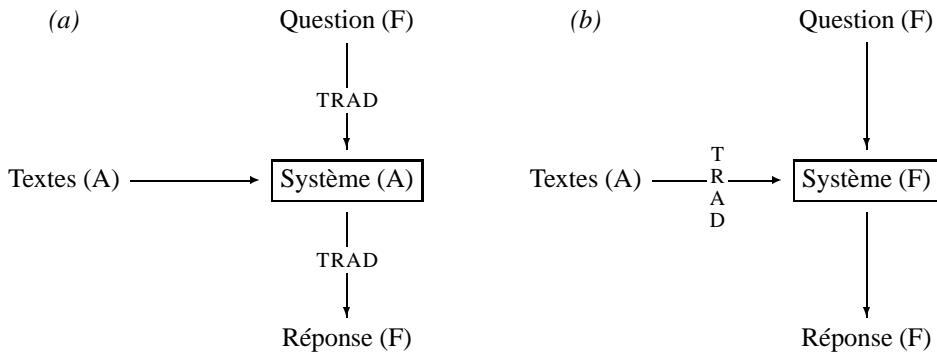


Figure 2: Deux approches pour l'adaptation d'un système monolingue anglais (A) pour répondre à des questions en français (F). En (a), le cœur du système demeure intact et des ressources linguistiques en anglais peuvent être utilisées. En (b), le cœur du système est transposé dans l'autre langue et les textes sont traduits.

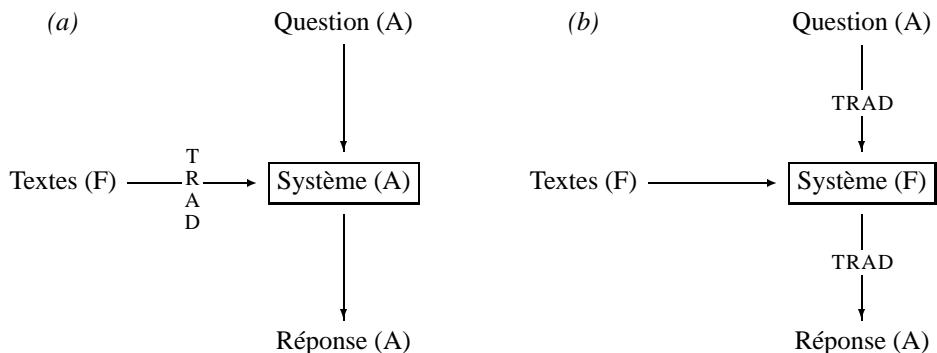


Figure 3: Deux approches pour l'adaptation d'un système monolingue anglais (A) pour fouiller des textes en français (F). En (a), le cœur du système demeure intact mais les textes doivent être traduits. En (b), le cœur du système est transposé dans l'autre langue.

fois avantageux d'*ouvrir* le système afin de tirer parti d'un maximum d'informations pouvant faciliter le processus de traduction. Cela permet entre autres de contourner le problème complexe de traduction de la question en une question anglaise syntaxiquement bien formée. Qui plus est, des techniques de traduction différentes peuvent être employées selon la tâche.

Pour ce faire, nous avons d'abord remplacé le module d'analyse de la question afin que l'analyse se fasse en français (figure 1b) car les méthodes de traduction probabilistes dont nous disposons sont peu performantes en regard de l'ordonnancement syntaxique des constituants de la phrase et les règles du module d'analyse ne pourraient s'appliquer adéquatement. La fonction d'extraction choisie par le module d'analyse peut être transmise directement au module d'extraction; seul le focus, lorsque la fonction d'extraction en requiert un, nécessite d'être traduit vers l'anglais pour les besoins du module d'extraction (entre autres parce que le focus doit être situé dans l'ontologie de Wordnet). Quant au module de recherche de passages, il requiert que les termes-clés de la question soient traduits vers l'anglais. Le module d'extraction de la réponse, lui, ne nécessite aucune modification. Examinons ces considérations en détail.

4.1 Conversion du module d'analyse des questions et traduction du focus

L'analyse des questions par Quantum se fait à l'aide d'expressions régulières combinant mots et étiquettes grammaticales. Le module anglais comprend environ 60 règles; nous en avons écrit à peu près le même nombre pour le français afin d'obtenir une couverture que nous espérions équivalente (nous présenterons une évaluation du module plus loin).

Pour écrire les règles d'analyse, nous avons suivi la même méthodologie pour l'anglais et le français: nous sommes partis du nombre limité de pronoms et adjectifs interrogatifs propres à la langue et nous avons recensé les constructions syntaxiques qu'ils commandent. Le français s'est avéré plus difficile à analyser car nous avons noté une plus grande flexibilité dans la formulation des questions. Par exemple, *How much does one ton of cement cost* peut se dire de deux façons en français: *Combien coûte une tonne de ciment* ou *Combien une tonne de ciment coûte-t-elle*. De plus, les pronoms interrogatifs — à la base de l'analyse — ne se traduisent pas nécessairement de la même façon: c'est le cas de *what* qui peut se traduire par *qu'est-ce que* dans *What is leukemia / Qu'est-ce que la leucémie*, par *que* dans *What does target heart rate mean / Que signifie rythme cardiaque cible* et par *quoi* dans *Italy is the largest producer of what / L'Italie est le plus grand producteur de quoi*. L'adjectif interrogatif *which*, comme dans *which city*, peut quant à lui prendre la marque du féminin et du pluriel: *quel, quels, quelle, quelles*. À ces difficultés s'ajoutent d'autres particularités de la langue française: le *t* euphonique de la forme interrogative de certains verbes à la 3^e personne du singulier (*Combien une tonne coûte-t-elle* mais pas *Combien deux tonnes coûtent-elles*), l'élation (*Qu'appelle-t-on*) et deux formes de passé (*Quand le téléphone fut-il inventé / Quand le téléphone a-t-il été inventé* alors que seule la forme *When was the telephone invented* est acceptable en anglais). De plus, les questions similaires à *Dans quelle série télévisée Pierce Brosnan a-t-il joué* présentent une difficulté au niveau de la segmentation des groupes nominaux. Il est en effet difficile de segmenter correctement *série télévisée* et *Pierce Brosnan*, alors qu'en anglais ces deux groupes nominaux sont la plupart du temps séparés par un auxiliaire: *What TV series did Pierce Brosnan play in*.

En plus de déterminer la fonction d'extraction à utiliser, les règles d'analyse servent aussi à identifier le focus. La nature sémantique du focus a parfois une influence sur le type de réponse attendu. C'est le cas de la question *Quel auteur a écrit sous le nom de plume "Boz"*, dont le focus *auteur* indique que la réponse doit être un nom de personne. Pour effectuer ce lien, nous consultons l'ontologie de Wordnet. Le focus doit donc être traduit en anglais avant que l'analyse de la question puisse être définitive. Pour ce faire, nous utilisons un moteur de traduction probabiliste IBM2 entraîné sur un ensemble de textes parallèles composés de débats de la Chambre des communes du Canada, de bulletins d'information de l'*Union européenne en ligne* et d'un échantillon de questions de TREC. Le modèle IBM2 tient compte de la position des mots dans la phrase source, ce qui permet de déterminer la traduction la plus probable compte tenu du focus et, dans une moindre mesure, des autres mots de la phrase source. Nous ne conservons que la meilleure traduction qui soit un nom.

Pour comparer les modules monolingue et bilingue, nous les testons sur un ensemble de questions tirées des campagnes TREC. Les 4 campagnes TREC-8 à TREC-11 ont donné lieu à la création d'un corpus de 1893 questions, chacune accompagnée des réponses trouvées dans la collection de textes par les systèmes participants et jugées correctes par des juges. Nous avons fait traduire ces 1893 questions en français afin de tester la version bilingue du système. Ce corpus de questions bilingue est disponible sur notre site web². Pour bâtir notre ensemble de test,

²www-rali.iro.umontreal.ca/ProjetLUB.fr.html

nous avons éliminé les 200 questions de TREC-8 à cause de la collection de textes différente utilisée lors de cette campagne. Des 1693 questions restantes, 114 n'ont pas de réponses, soit parce que les questions ont été éliminées des campagnes pour causes d'irrégularités, soit parce qu'elles n'ont pas de réponse connue dans la collection de textes. Nous avons sélectionné au hasard 789 questions parmi les 1579 restantes, c'est-à-dire la moitié.

Nous avons mesuré que les expressions régulières (utilisées en conjonction avec l'ontologie de Wordnet) permettent de choisir la bonne fonction d'extraction pour 96 % des questions en anglais. Cette performance diminue à 77 % pour les questions en français. La dégradation de 20 % par rapport au système monolingue est due autant à la couverture des expressions régulières qu'à la traduction du focus. Le focus est traduit de façon satisfaisante une fois sur deux; la cause d'erreur principale est l'absence du mot à traduire dans le corpus d'entraînement. Notamment, les demandes de définition portent sur des expressions spécialisées: *Qu'est-ce que la thalassémie, l'amoxicilline, un shaman, etc.* Ces expressions étant inconnues du moteur de traduction et la recherche de définitions s'appuyant principalement sur la recherche d'hypéronymes du focus, il s'avère impossible d'utiliser Wordnet pour déterminer que les hypéronymes *anemia, antibiotic* et *spiritual leader* rencontrés dans le texte constituent des réponses correctes.

De la même façon, la recherche d'un hyponyme du focus pour répondre à une question de spécialisation du genre *Quelle fleur Vincent Van Gogh a-t-il peint? → le tournesol* est sérieusement compromise si le focus *fleur* est faussement traduit par *horse* ou si le focus est tout simplement inconnu du moteur de traduction. Mais dans ce cas-ci, c'est l'analyse de la question qui a fait échouer le repérage de la réponse car le focus a été mal identifié dès le départ, à cause d'une erreur de segmentation de *fleur Vincent Van Gogh* en deux groupes nominaux.

4.2 Traduction des termes-clés pour la recherche de passages

Indépendamment du domaine de la QR, beaucoup de chercheurs se sont penchés sur le problème de la recherche d'information translinguistique. La tendance est actuellement de fondre le modèle de traduction avec le modèle de recherche d'information (Kraaij *et al.*, 2003). Cependant, étant donné que notre moteur Okapi ne nous permet pas de modifier le modèle de recherche, nous avons opté pour l'approche classique: d'abord obtenir des termes-clés par traduction et ensuite s'en servir comme requête pour la recherche de passages.

Premièrement, nous utilisons un moteur de traduction IBM1 pour obtenir les termes ayant la plus forte probabilité de faire partie d'une traduction de la question. À la différence du modèle IBM2 utilisé pour traduire le focus, le modèle IBM1 ne tient pas compte de la position des mots de la question: chacun des mots de la question contribue de façon égale à produire les termes en français les plus probables. Les termes générés et qui ne sont pas vides de sens constituent les termes-clés de la requête. Nos expériences ont montré que la recherche translinguistique est plus fructueuse lorsque la requête en français contient autant de termes-clés que la question en anglais en contient, c'est-à-dire en moyenne 5.

Nous avons testé le module de recherche d'information translinguistique avec le même ensemble de questions que pour le module d'analyse des questions. La recherche s'est effectuée dans la collection de textes propre à TREC-9/TREC-10 ou à TREC-11, selon la question. La performance est mesurée à l'aide de la précision moyenne (Kraaij *et al.*, 2003). Le module de recherche monolingue obtient une précision moyenne de 0,570 avec les questions originales en

anglais et le module translinguistique obtient une précision moyenne de 0,467 avec la version française des questions, soit une dégradation de 18 %. À la différence de la traduction du focus, la traduction erronée d'un terme-clé ne compromet pas la recherche de la réponse, surtout lorsque la question est longue et que les termes-clés sont nombreux, comme dans *Quel était le nom de la comédie télévisée dans laquelle Alyssa Milano jouait auprès de Tony Danza?*

4.3 Traduction de la réponse

Une fois la réponse repérée dans un texte, elle peut ou non être traduite, selon l'usage auquel le système est destiné. La traduction de la réponse de l'anglais vers le français fait partie de nos objectifs bien que nous ne nous y soyons pas encore attaqués. Le contexte particulier de la QR joue ici en notre faveur: en effet, beaucoup de réponses sont des entités nommées qui ne requièrent pas de traduction. Par exemple, sur un échantillon de 200 questions prises au hasard, 25 % ont pour réponse un nom de personne ou de lieu identique dans les deux langues, un nombre, une date, une raison sociale ou un titre d'ouvrage. Pour les autres types de réponse, il serait intéressant d'utiliser la question pour désambiguer les termes avant de les traduire.

4.4 Performance du système complet

Puisque les réponses données par la version bilingue de Quantum sont pour l'instant en anglais, il est possible d'utiliser la procédure d'évaluation automatique fournie lors des campagnes TREC. Le système bilingue, testé sur la version française des questions de notre ensemble de test, obtient 44 % moins de bonnes réponses que le système monolingue testé sur les questions originales. Nous avons comparé avec une alternative plus simple consistant à traduire les questions du français vers l'anglais à l'aide d'un moteur de traduction tel que Babelfish³ puis à utiliser le système monolingue, mais cette approche a entraîné une perte de performance de 53 %, donc pire que celle du système bilingue que nous avons développé.

5 Conclusion

Nous avons montré comment il est possible de transformer un système de question-réponse monolingue en un système bilingue capable de répondre à des questions posées dans une langue différente de celle de la collection de textes. Théoriquement, les modifications peuvent se faire sans toucher au système lui-même par la simple traduction des entrées/sorties. Cependant, tant que les techniques de traduction ne permettront pas d'obtenir des traductions parfaites, il peut s'avérer plus profitable de scinder le problème et de sélectionner différents points de traduction à l'intérieur du système. Dans le cas de notre système Quantum, nous avons utilisé le modèle IBM1 pour obtenir, à partir de la question, des termes-clés pour la recherche de passages pertinents. Nous avons utilisé le modèle IBM2 pour traduire le focus des questions et nous avons écrit de nouvelles règles d'analyse en français dans le but d'éviter d'avoir à produire une traduction complète et bien formée de la question, sur laquelle les règles d'analyse en anglais auraient pu s'appliquer. Le tout a entraîné une dégradation de performance de 44 % par rapport au système monolingue d'origine.

³world.altavista.com

Nous espérons qu'un système bilingue anglais/français donnera aux francophones l'accès à un plus large éventail de sources d'information. Nous estimons qu'actuellement, un francophone qui utilise la version bilingue de Quantum sur une collection de textes en anglais ou qui utilise un système monolingue français sur une collection en français réduite est confronté au même manque à gagner. En effet, dans le premier cas, la bilinguisation du système monolingue a entraîné une dégradation de performance de 44 %. Dans le deuxième cas, pour évaluer le manque à gagner entraîné par la documentation réduite disponible en français, nous avons testé la version monolingue anglaise de Quantum sur une collection de textes dix fois plus petite que la collection originale et nous avons observé une perte de réponses correctes de l'ordre de 43 %. Cependant, à mesure que les moteurs de traduction s'amélioreront, nous sommes confiants que la QR multilingue offrira des performances de plus en plus intéressantes et ce, sans compter l'effet combiné de l'accès à des collections de textes en français *et* en anglais.

Remerciements

Ce projet a été soutenu financièrement par les Laboratoires Universitaires Bell (LUB), le Conseil de recherche en sciences naturelles et en génie du Canada (CRSNG) et le Conseil national de recherches du Canada (CNRC). Nous désirons remercier plus particulièrement Guy Lapalme et les instigateurs du projet: Joel Martin, du CNRC, et Elliott Macklovitch, du RALI.

Références

- CLARKE C. L. A., CORMACK G. V., LASZLO M., LYNAM T. R. & TERRA E. L. (2002). The Impact of Corpus Size on Question Answering Performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research Information Retrieval (SIGIR 02)*, Tampere, Finlande.
- GRAESSER A., PERSON N. & HUBER J. (1992). *Mechanisms that Generate Questions*. Lawrence Erlbaum Associates: Hillsdale, New Jersey.
- KIM H. & SEO J. (2002). A Reliable Indexing Method for a Practical QA System. In *Proceedings of the COLING 2002 Post-conference Workshops*, Taipei, Taiwan. Présenté au "Workshop on Multilingual Summarization and Question Answering 2002".
- KRAAIJ W., NIE J.-Y. & SIMARD M. (2003). Embedding Web-based Statistical Translation Models in CLIR. *Computational Linguistics*, **29**(2). À paraître.
- MAGNINI B., NEGRI M., PREVETE R. & TANEV H. (2001). Multilingual Question/Answering: the DIOGENE System. In *Proceedings of TREC-2001*, Gaithersburg, Maryland.
- PLAMONDON L. (2002). Le système de question-réponse QUANTUM. Mémoire de maîtrise, Université de Montréal. www.iro.umontreal.ca/~plamond1.
- PLAMONDON L., LAPALME G. & KOSSEIM L. (2002). The QUANTUM Question Answering System at TREC-11. In *Notebook Proceedings of TREC-11*, Gaithersburg, Maryland.
- SEKI Y. & HARADA K. (2002). Summarization-Based Japanese Question and Answering System from Newspaper Articles. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas de Gran Canaria, Espagne: ELRA.
- VETULANI Z. (2002). Question Answering System for POLISH (POLINT) and its Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas de Gran Canaria, Espagne: ELRA.
- VOORHEES E. M. (2002). Overview of the TREC 2002 Question Answering Track. In *Notebook Proceedings of The Eleventh Text Retrieval Conference (TREC-11)*, Gaithersburg, Maryland.

Cross-Language Information Retrieval using Ontology

Ahmed Abdelali, James Cowie, David Farwell,
Bill Ogden, and Stephen Helmreich

Computing Research Laboratory
Box 30001/3CRL
New Mexico State University
Las Cruces, NM 88003 USA
[{ahmed}](mailto:{ahmed}@crl.nmsu.edu)@crl.nmsu.edu

Abstract

In this paper we present a description and an evaluation of ontology- based Cross-Language Information Retrieval. Earlier systems we have developed used bilingual dictionaries to support a user in selecting terms in the language of the documents being retrieved. This presents the user with the problem of deciding if the translations are the correct senses needed for the query. The system described here replaces the bilingual dictionaries by a pair of language-ontology lexicons. The user can see definitions of the senses in the ontology and then select matching terms in the target language. This allows better control of the generation of the new query in the target language.

Keywords

Cross-Language Information Retrieval, Unicode, Ontology, Knowledge Based Information Retrieval.

1 Introduction

The aim of Information Retrieval (IR) is to find and retrieve documents relevant to a given query, usually where documents and query are in the same language. With further advances in research and technology the goal was extended beyond language barriers to include different in different languages, which is known as Cross Language Information Retrieval (CLIR). Using the available capabilities of the Computing Research Laboratory at New Mexico State University, we developed a new approach and produced a cross-language retrieval system with meaning-based alternatives for query translation. This paper includes an overview of the approach, a description of the system, and a preliminary evaluation of performance.

2 Background

The knowledge-based approach described here was intended to solve problems that exist in corpus-based, bilingual-dictionary-based, and machine-translation-based Information Retrieval. For example, there are problems with query recall, problems with ambiguity, and problems matching the query to actual documents. We started this project by developing tools to support acquisition of knowledge for the ontology and lexicon resources. These tools included an editor by which acquirers could populate the ontology concepts, as well as an interface that allowed the lexicographers to acquire and expand the English, Spanish and Chinese lexicons. These tools also supported mapping of acquired lexemes to ontology concepts. After developing these resources, we used Keizai, the cross-language information retrieval system, developed during previous CRL projects (Davis & Dunning 1995, Ogden et al. 1999, Ogden & Davis 2000), and modified it to support ontology-based queries for all the three languages. The Unicode-based nature of the system facilitated the modification.

On the other hand, the Knowledge-based Ontology developed at CRL and used in the Mikrokosmos Project (Mahesh & Nirenburg 1995, Mahesh 1996) is language neutral and combines the information in a thesaurus with that in an encyclopedia. It also contains concept-level co-occurrence constraints. The ontology is connected to lexical elements of natural languages via a dictionary, either directly or through the English side of available bilingual dictionaries. The combination of ontological knowledge and its connection to the dictionaries gives the approach a powerful means for resolving IR problems. Through the ontology and its related lexicons/dictionaries the user of the IR system has the ability to do a direct lookup in any of the dictionaries. E.g., the English word “ship” leads to immediate instantiation of the corresponding ontological concept “SHIP”, and also to words in other languages, such as Spanish and Chinese, which are also used to express the concept “SHIP”. (Figure 1)

CONCEPT : SHIP
DEFINITION : any large vessel navigating deep waters
IS-A : SURFACE-WATER-VEHICLE
SUB : OIL-TANKERSAILING-SHIPTRAWLERWARSHIP
ENGLISH : aircraft-N3brig-N2brim-N2broadside-N2craft-N1cutter-N1derelict-N2draft-N11flagship-N2fleet-N2flotilla-N2freighter-N1galley-N1galley-N3icebreaker-N1ketch-N1liner-N1minesweeper-N1sailing vessel-N1ship-N1ship-N2shipwreck-N2tender-N3vessel-N1wreck-N3
SPANISH : - barco-N1 barco-N3 bergantín-N1 borde-N6 bote-N6 buque insignia-N1 calado-N1 carguero-N1 costado-N2 cíuter-N1 flotilla-N2 galera-N1 nave-N1 nave-N2 navio-N1 navío-N1 navío-N2
CHINESE : - 冷藏船-N 散货船-N 汽车运输船-N 石油液化气船-N 船-N 船型-N 船舶-N 货轮-N 远洋船-N 集装箱船-N

Figure 1: Ontological Concept “SHIP” and related natural language entries

Every ontological concept contains a set of features that allow the user to disambiguate the concept from other hyponyms; also the concept, through a set of defined relations, is connected to other concepts. The latter serves as a means for expansion of the IR query.

To evaluate the performance of the system we performed experiments that would explore the advantages and problems with the approach. The evaluation was conducted in a comparative fashion, as an evaluation of the ontology-based versus dictionary-based approaches to information retrieval.

3 Tool Description

The system is based on Keizai, a cross-language, interactive, retrieval and summarization system that uses URSA (Unicode Retrieval System Architecture) and MINDS (Multilingual Interactive Document Summarization), developed at CRL. Keizai uses a combination of automatic and user-assisted methods to build and improve cross-language queries. It sends the modified queries to language-specific query modules to retrieve documents, and displays various types of English summaries of the retrieved document (See Figures 2, 3 and 4).

Figures 2, 3, and 4 illustrate the steps taken in retrieving documents in Chinese containing the Chinese equivalents of the word “ship”. In this task, the user enters the English word in the English Query Interface-Interactive Selection. The result of the request will be the set of matches in a set of bilingual dictionaries; the entries are sorted by language. The user then chooses the closest translations that could match the original query. In the final step after constructing the new query in the target language, the system will return a set of documents relevant to the query. The user has also the possibility of translating the returned document back to English. Depending on the source language, we either use an internal translation system MEAT (Amstrup et al. 2000, Zajac et al. 2001) (Chinese) or an external translation engine Systran (Spanish).

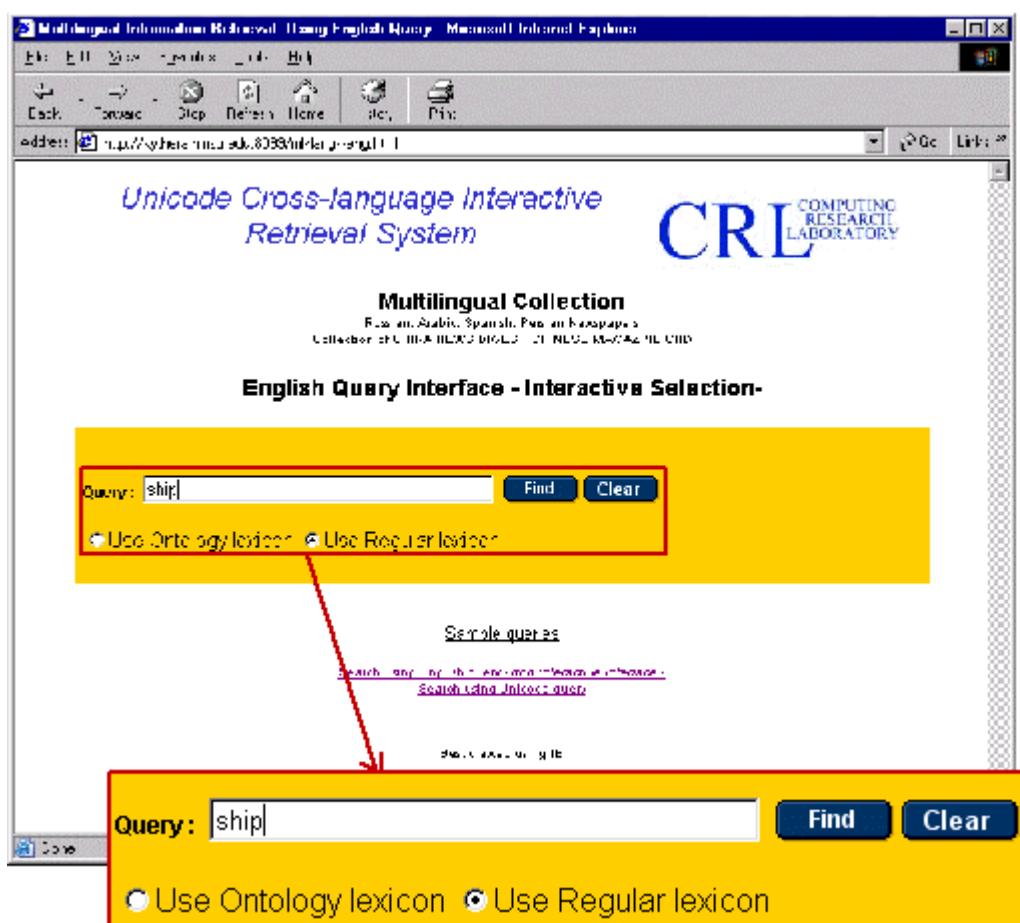


Figure 2 : Query on Keizai using regular lexicon

The screenshot shows two windows of the "Indexed Unicode Archive Resource". The top window is a browser-based interface with tabs for "Actions", "New Search", "Generated Unicode Query", and "Matches Retrieved". The bottom window is a native application window titled "Indexed Unicode Archive Resource" with a search bar containing "ship", a "Translate" button, and a "New Search" link. It also has a "Generated Unicode Query" field, a "Clear" button, and a "Retrieve" button. A red arrow points from the "Matches Retrieved" section of the bottom window to the "Matches Retrieved" section of the top window.

Indexed Unicode Archive Resource:

ship

Translate

New Search

Generated Unicode Query : Stem Search

Clear Retrieve

Matches Retrieved

1.	1. <u>غاطس</u> draft of a ship	
2.	2. <u>سفن</u> ship	
3.	3. <u>سفن</u> ship	
4.	4. <u>مساکب</u> ship	
5.	5. 船无线电寂静时间 ship radio silence	

Figure 3 : Chinese and Arabic “ship” using regular lexicon

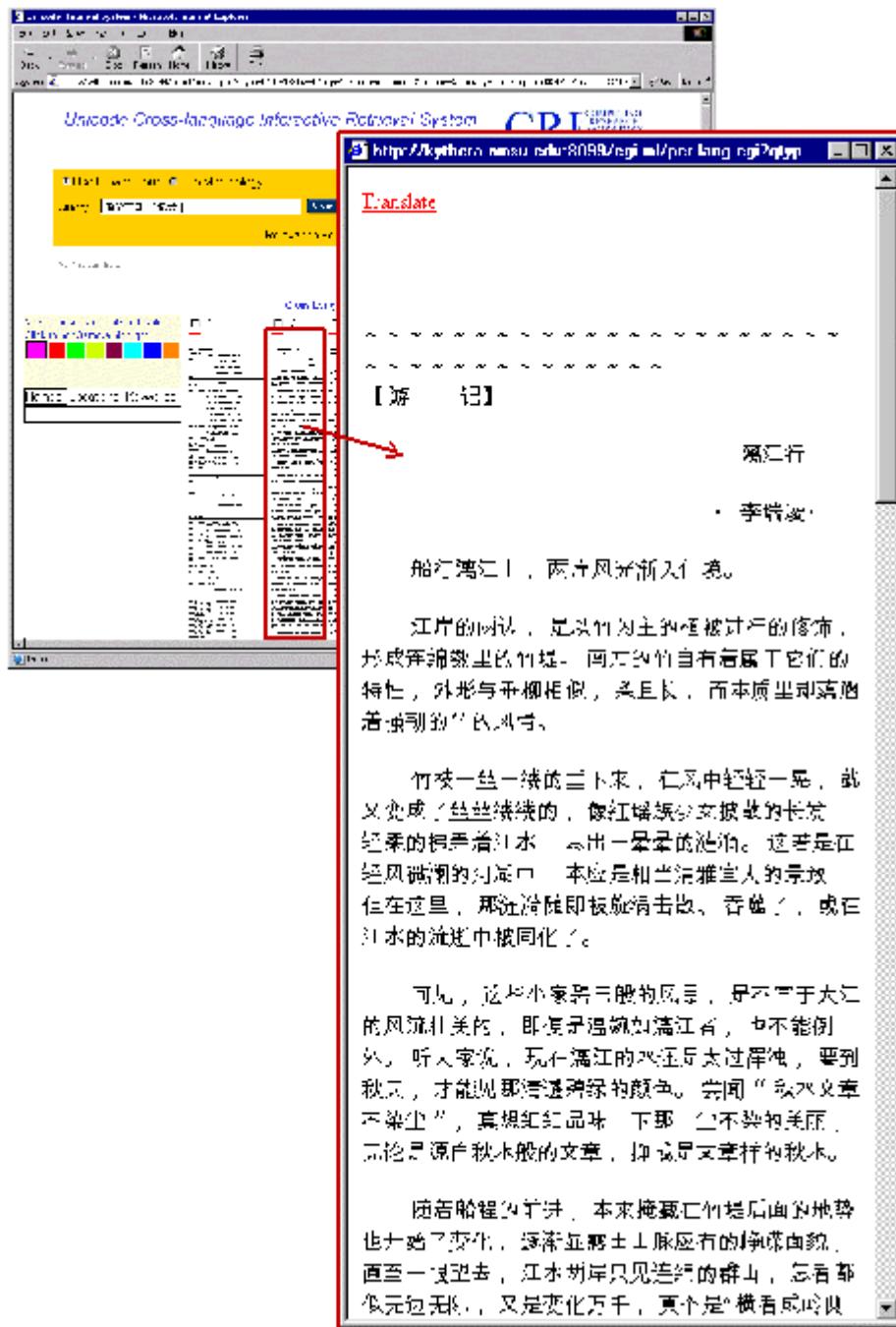


Figure 4 : Chinese text retrieved by Keizai using regular lexicon

The new approach using the ontology gives the user another route to retrieve the data. Since the ontology is connected to different lexicons available via semantic dictionaries, the interface provides a wider variety of lexical choices in the target languages, but these are organized by concept. Figure 5 illustrates a query for the word “building” using the ontology lexicon. The search outputs a list of ontology concepts containing the word “building”, each for a different sense of the word. As shown in Figure 6, the user selects the intended meaning of the word. Then the editor shows that ontology entry for “building”, in the right frame, with one or more English and Spanish equivalents of the word mapped to it. The user then selects one or more appropriate equivalents, in this case “construcción” and enters it in the query automatically (Figure 6).

Cross-Language Information Retrieval using Ontology

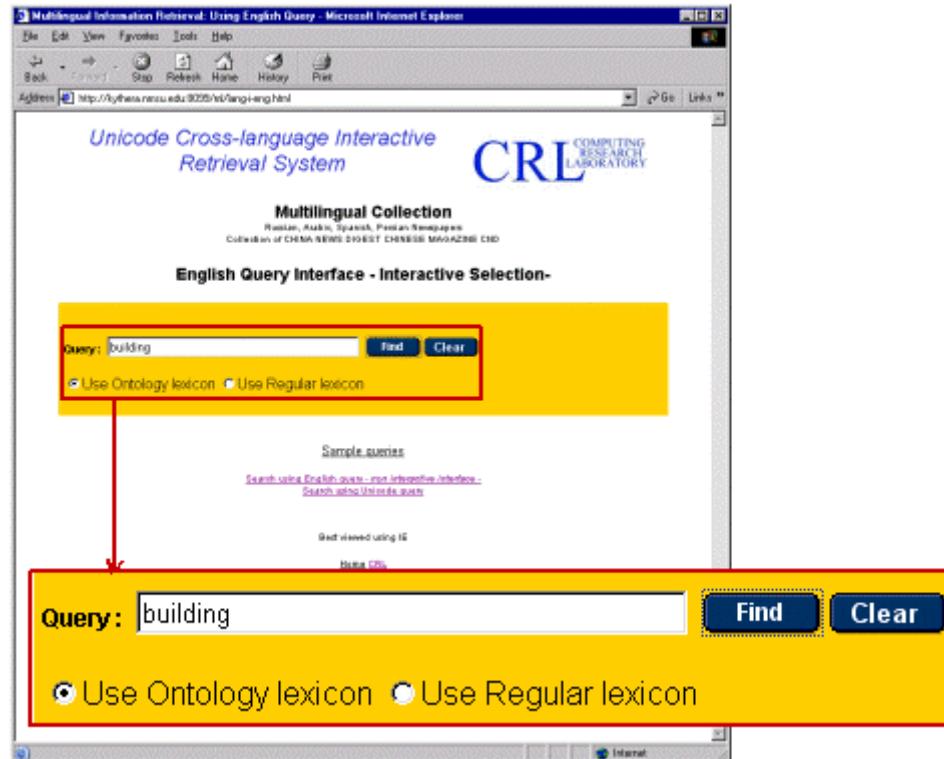


Figure 5 : Using ontology lexicon through the new Keizai interface

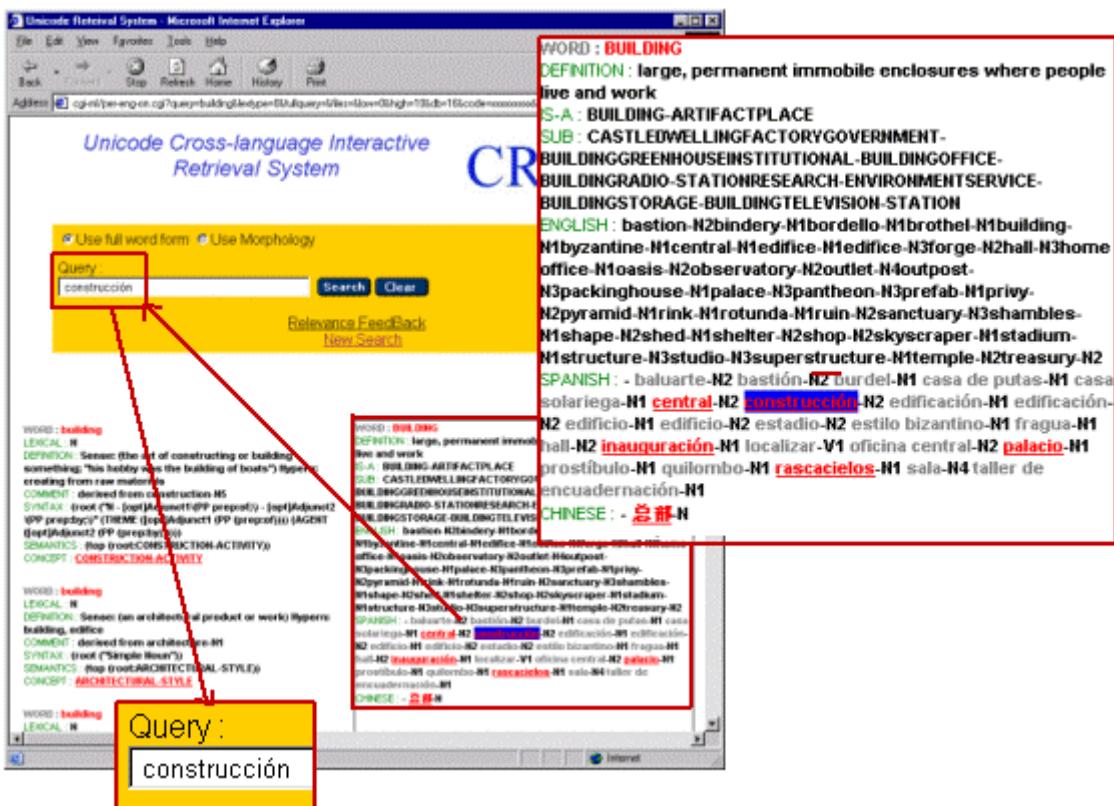


Figure 6 : Spanish equivalent for “building” selected for the new query

Keizai retrieves a list of Spanish texts that contain the word “construcción” and outputs them in thumbnail form. The user then can click on each of the texts retrieved for further analysis.

As a visual aid, the occurrences of a keyword in the texts retrieved can be highlighted by selecting color options on the interface and mouse-over action on the keyword.

4 Evaluation

Evaluation was carried out over two versions of the IR system, one using the ontology versus one using the regular lexicon. We chose Chinese as the target language for this test. For each test, we took the first 20 documents from the results for comparison. The testing procedures are as follows:

Query	Method	Words Available	Words Selected	Total retrieved documents	Relevant document in the first 20
Buy	Ontology	27	17	352	20
	Regular	10	5	316	20
Money	Ontology	40	17	413	20
	Regular	21	8	324	20
Market	Ontology	10	10	728	20
	Regular	22	7	1256	20
Guest	Ontology	2	2	488	13
	Regular	6	3	218	19
Discover	Ontology	1	1	21	10
	Regular	7	2	1430	19
Share	Ontology	8	8	177	20
	Regular	199	2	16	4

Figure 7 : Table comparing results of Ontology versus regular lexicon IR

Information retrieval using the ontology:

1. An English query is entered;
2. "Use Ontology lexicon" option is selected to generate the list of concepts;
3. One concept from the list is chosen;
4. All related Chinese words connected to each concept are selected and specified as Chinese queries;

5. The retrieval results are displayed.
6. The first 20 documents are checked against the queries for relevancy.

Information retrieval using regular lexicon:

1. An English query is entered;
2. "Use Regular lexicon" option is selected to generate the list of Chinese-English pairs;
3. All related Chinese words in the list are selected and entered as Chinese queries;
4. The retrieval results are displayed.
5. The first 20 documents are checked against the queries for relevancy.

Figure 7 shows the results.

5 Discussion

The purpose of ontology-based IR is to narrow down the search by eliminating the number of meanings (sense/concepts) of the query. Only a relevant concept is chosen to find the Chinese words. The user selects the Chinese words as queries for retrieval. To reach high-quality results the following supporting components are necessary:

- An ontology with wide range of coverage in world knowledge;
- A large-size lexicon in Chinese, English and Spanish with accurate mapping to the ontology concepts.
- A large corpus of on-line documents in three languages.
- A sophisticated IR strategy.

Even with the above supports, ontology IR only controls the meaning of the English query. Once the Chinese words are found, there is no control of the Chinese queries. That is, each Chinese word may have several meanings that result in irrelevant documents being selected.

The ontology-base IR approach relies heavily on the ontology, and particularly on the accuracy of the lexicon mapping in various languages. In case no appropriate concept exists or if the constraints on the concept are neglected, the translated query can be far different than the original meaning of the English query. For example, in the current ontology there are 832 English words mapping to the concept OBJECT and 787 English words mapping to the concept EVENT. This will generate a Chinese query with a very general sense. On the other hand, there are 968 Spanish words mapping to OBJECT and 921 Spanish words mapping to EVENT. This will result in unmanageable Spanish IR. Much work is needed in building high-quality lexicons in three languages.

The lexicon mapping format needs to be consistent. Currently there are two different ways of mapping a lexical item to a PROPERTY (as opposed to an EVENT or an OBJECT):

Case 1. mapping a noun to a PROPERTY, such "price"

COST[DOMAIN COMMODITY]

COMMODITY[DOMAIN-OF COST]

The IR system extracts COST as a concept in the first case and COMMODITY in the second case that results in wrong query.

Case 2. mapping an adjective to a PROPERTY, such as "equal" or "equality"

EQUAL[DOMAIN OBJECT]

OBJECT[DOMAIN-OF EQUAL-TO]

The IR system extracts EQUAL as a concept in the first case and OBJECT in the second case, resulting in wrong query. Since the lexicon files are shared with various projects for different purposes, the lexicon-mapping algorithm cannot be based only on IR needs. Therefore different strategies of concept extraction must be taken into account.

Problems particularly in Chinese: Because Chinese texts are presented as a sequence of characters, without segmentation, there is no way to indicate word boundaries. Efficient segmentation may be needed to improve the IR results.

Problem in the method of extracting concepts: Currently the system only extracts the head concept regardless of constraints. As result the concept is generalize, so that, for example, 'man', 'woman', 'child' are all mapped to HUMAN. Therefore the translated query is too general in comparison to the original English query.

The system also is not aware that the meaning of adjective is represented in the constraint. In most case the translation of adjective queries are incorrect. The case is similar when mapping a noun to PROPERTY.

As a result, irrelevant documents are sometimes collected.

The current Chinese lexicon is too small and in a specific domain, while the Chinese corpus collection is in the general domain. This fact limits the number of documents retrieved. The Chinese lexicon needs to be extended.

English lexicon needs to be checked with respect to concept accuracy and format consistency, in order to reduce the number of irrelevant documents.

6 Conclusion and future work

In this simple attempt to use a new approach for replacing conventional lexically-based IR with ontology-based IR we demonstrated that the ontology-based IR performed equivalently to the regular lexicon IR. Improving the quality and size of the ontology could improve results. Promising results could be achieved with little effort by fixing the ontology inconsistencies and populating the attached lexicons. Another point to consider for future work to improve the system includes correcting the wrong concept mapping in both English and Spanish and changing method of extracting concepts from adjectives.

Acknowledgments

Keizai was originally developed by Mark Davis. The principal designers of the Mikrokosmos Ontology are Sergei Nirenburg and Victor Raskin.

References

Davis, Mark, and Ted Dunning. (1995) Cross-Language Text Retrieval using Evolutionary Optimization. (EP95 in San Diego).

Amtrup, Jan W., Hamid Mansouri Rad, Karine Megerdoomian, and Rémi Zajac. (2000) Persian-English Machine Translation: An Overview of the Shiraz Project. NMSU CRL Technical Report. MCCS-00-319

Mahesh, Kavi, and Sergei Nirenburg. (1995). Semantic Classification for Practical Natural Language Processing. Proceedings of the 6th ASIS SIG/CR Classification Research Workshop: An Interdisciplinary Meeting. Chicago, Illinois.

Mahesh, Kavi. (1996). Ontology Development for Machine Translation: Ideology and Methodology. NMSU CRL Technical Report. MCCS-96-292.

Ogden, William, James Cowie, Mark Davis, Eugene Ludovik, Sergei Nirenburg, Hugo Molina-Salgado, and Nigel Sharples. (1999) Keizai: An Interactive Cross-Language Text Retrieval System. Paper presented at the Workshop on Machine Translation for Cross-language Information Retrieval, Machine Translation Summit VII, September 13-17, 1999, Singapore.

Ogden, William, and Mark Davis. (2000) Improving Cross-Language Text Retrieval with Human Interactions. Hawaii International Conference on System Sciences, HICSS-33 January 4-7, 2000.

Zajac, Rémi, Ahmed Malki, Ahmed Abdelali, James Cowie, and William Ogden W. (2001). Arabic-English NLP at CRL, Proceedings of the Arabic NLP Workshop ACL/EACL in July 2001, Toulouse (France).

Repérage de traduction et commutation interlingue : Intérêt et méthodes

Olivier Kraif

LIDILEM

Laboratoire de Linguistique et de Didactique des Langues Etrangères et
Maternelles

Université Stendhal - Grenoble 3
38400 Saint-Martin d'Hères – FRANCE
Olivier.Kraif@u-grenoble3.fr

Résumé – Abstract

Le développement des corpus multilingues alignés a rendu possible la formalisation, et la systématisation, d'une forme originale d'observation, à savoir le *repérage de traduction*. Dans un premier temps, nous montrons comment ce type de repérage fournit des critères intéressants pour l'identification d'unités polylexicales, pour l'explicitation du sens des lexies, et plus généralement pour l'objectivation de structures sémantiques complexes telles que la polysémie. Afin de formaliser la mise en oeuvre de ce type de repérage nous examinons le test de commutation interlingue. Enfin, nous montrons comment ce test peut-être étendu grâce aux techniques d'extraction de correspondance lexicales, qui permettent de faire apparaître, à partir d'observations quantitatives, des régularités pertinentes découlant de phénomènes contrastifs.

The development of multilingual aligned corpora has made possible the systematization of an interesting kind of observation, namely the *translation spotting*. We first describe how translation spotting can be useful in grasping multi-word compounds, specifying the denotation of lexical units, and making easier the objective observation of semantic phenomena such as synonymy or polysemy. As a criterion, we propose, and discuss, an interlingual commutation test. We then suggest that it is worth applying it at a massive scale, because contrastive phenomena appear at a statistical level with the emergence of regularities.

Keywords – Mots Clés

Repérage de traduction, commutation interlingue, traduction, multi-texte, alignement multilingue, lexicologie
Translation spotting, interlingual commutation, translation, multi-text, multilingual aligning, lexicology.

1 Introduction

Avec le développement croissant des corpus textuels sous format numérique, on assiste à un regain d'intérêt pour la linguistique de corpus, et de nouveaux objets naissent en même temps que de nouvelles méthodes : le corpus n'est plus seulement le lieu où s'exerce avec acuité le regard du linguiste, dans une démarche de description, d'explicitation et d'analyse, c'est aussi un réservoir de phénomènes de masse, d'informations statistiques destinées à alimenter des modèles numériques, pour la mise en œuvre d'outils informatiques consacrés au traitement du langage.

Fruits de ce développement, les multi-textes occupent une position originale dans la famille des corpus numériques, car ils relèvent d'une activité communicative complexe et peu étudiée par les linguistes, la traduction, et concernent plusieurs langues. Ces corpus multilingues, qui rassemblent des textes traduits dont les portions équivalentes sont alignées à des niveaux de granularité plus ou moins fin (paragraphe, phrase, mot), servent d'abord des objectifs pratiques, comme l'aide à la traduction, la lexicographie bilingue, ou l'enrichissement de modèles statistiques destinés à la traduction automatique. Mais les multi-textes constituent aussi des objets intéressants du point de vue de la description linguistique.

Dans l'étude et le traitement de ces corpus multilingues, un nouveau type d'observation a pu être formalisé, grâce au traitement automatique : le *repérage de traduction (lexical spotting)*, pour reprendre le terme utilisé lors de la seconde campagne d'évaluation Arcade. Cette tâche consiste simplement à déterminer, étant donné une unité du texte source, quelle est l'unité ou l'expression équivalente. Or, la possibilité de traduire une expression par une ou plusieurs expressions équivalentes dans la langue d'arrivée, donne des indices précieux sur le sens de cette unité, dans son contexte d'occurrence. Les phénomènes de polysémie ou de figement sémantique ne manquent pas de se manifester à ce niveau, car comme le disait Greimas (1970), la traduction est l'amorce d'une explicitation du sens. Ainsi, d'après nous, le repérage de traduction est révélateur d'une grande variété de phénomènes linguistiques et contrastifs. Dans un premier temps, nous tenterons d'en préciser l'étendue. Nous tâcherons ensuite d'exposer les méthodes, manuelles ou automatiques, qui peuvent s'appliquer à ce type de dépouillement. Notamment, nous examinerons et discuterons le test de commutation interlingue, inspiré de Catford (1965) et Mahimon (1999). Enfin, nous montrerons que le principe de la commutation peut être généralisé si l'on se situe au niveau des régularités qui émergent de la masse des phénomènes. Nous tenterons alors d'indiquer comment de nouveaux outils, appliqués à cet objet linguistique encore méconnu, le multi-texte, permettent d'étendre le champ de l'observation linguistique.

2 Le repérage de traduction

Considérons l'exemple suivant (tiré du corpus JOC¹) :

Fr. : Eu égard à l'intention de la Commission de présenter un Livre vert sur le secteur des postes dans la Communauté

Angl. : Having regard to the Commission's intention to issue a Green Paper on the postal sector in the Community;

¹ Le corpus JOC, utilisé dans le projet ARCADE, est constitué de questions écrites soumises à la Commission européennes en 1993, dans les Séries C du Journal officiel de la Communauté européenne, et collectées dans le cadre du projet MLCC-MULTEXT (<http://www.lpl.univ-aix.fr/projects/multext/CORP/JOC.html>).

D'une façon intuitive, on peut en tirer un certains nombres de correspondances :

(Eu égard à ; Having regard to), (Commission ; Commission), (intention ; intention), (présenter ; to issue), (Livre vert ; Green Paper), (secteur des postes ; postal sector), (Communauté ; Community).

Ce type de repérage de traduction (pour lequel nous fournirons plus loin des critères), fait apparaître différentes sortes d'informations :

- Le long de l'axe syntagmatique, d'une part, il aboutit à une segmentation spécifique des unités. Certaines de ces unités sont directement issues de ce qu'on pourrait appeler la non-compositionnalité traductionnelle, pour reprendre une notion introduite par Isabelle (1992). Par exemple, *Livre vert* et *Green Paper* doivent être appariées en bloc, car la relation d'équivalence ne se décompose pas au niveau des formes qui les constituent. Ainsi, le repérage de traduction peut fournir un critère pour l'extraction de certaines expressions figées. Ce critère peut être intégré à des méthodes quantitatives, comme l'a montré Melamed (1997b), qui note que pour des unités non compositionnelles, la mesure d'information mutuelle entre unités source et cible est supérieure quand on considère ces unités d'un bloc. Notons que la non-compositionnalité connaît des degrés, et qu'elle peut se manifester avec moins de force au niveau de divergences mineures. Par exemple, le complément nominal *des postes* est traduit par l'adjectif relationnel *postal*. Cette divergence nous a conduit à traiter ces syntagmes en bloc, afin de faire correspondre des unités homogènes. Or, l'examen des occurrences sur l'ensemble du corpus JOC indique que *postal sector* est en relation avec *secteurs des postes* dans 5 cas sur 7 et avec *secteur postal* dans les 2 cas restants, confirmant l'hypothèse d'un emploi préférentiel de la première combinaison. Ainsi, les divergences observées lors du repérage de traduction peuvent indiquer des usages qui auraient peut-être échappé à l'examen monolingue. C'est le cas pour de nombreuses collocations : l'impossibilité de les traduire mot à mot est révélatrice de leur degré de cohésion. Elles sont certes observables dans un corpus monolingue, de par leur récurrence, mais elles sont plus facilement repérables dans un corpus aligné. Ainsi, ce que le repérage de traduction fait apparaître, c'est un niveau de segmentation propre au plan contrastif, qui définit des « unités de traduction » au sens de Vinay et Darbelnet (1959 : 37). Comme le note Véronis (2000) dans la perspective de l'alignement, le repérage monolingue des unités n'est pas indépendant de leur mise en correspondance : « la détermination des unités dans la langue source est dépendante de langue cible (par exemple, il faut aligner d'un bloc *demande de brevet* et *Patentanmeldung* [BLANK 2000] alors que l'alignement peut se fractionner avec *domanda di brevetto*). » Ces unités de traductions sont intéressantes à deux niveaux : d'une part, elles peuvent révéler l'existence d'une unité phraséologique pertinente au niveau de l'idiome ; d'autre part, capitalisées, elles peuvent intervenir dans le processus de traduction afin d'effectuer un transcodage plus modulaire des unités.

- Sur le plan paradigmatique, on observe une liste d'unités qui portent, dans ce contexte précis, des sens équivalents. Ainsi *intention* est traduit par son cognat *intention*, *Livre vert* est traduit par *Green Paper*, *Eu égard à* par *Having regard to* : ce type de relation, observable en de nombreux points du corpus, est capital pour le traducteur, le lexicographe ou le terminologue. En ce sens, le repérage de traduction constitue une étape préalable à la constitution d'un dictionnaire (général ou terminologique) bilingue. Notons que la correspondance des unités peut dépasser le strict niveau lexical : rien n'empêche de s'intéresser au repérage de morphèmes, ou de traits grammaticaux. L'étude sur corpus permet alors d'observer des régularités concernant l'équivalence d'unités à valeur grammaticale.

Outre ce type de relation biunivoque, l'accumulation d'un grand nombre d'observations issues du repérage de traduction permet de faire affleurer des structures complexes relatives à l'organisation sémantique des unités. Par exemple, la polysémie d'une unité en langue source peut être manifestée par sa mise en correspondance avec des unités cibles appartenant à des champs sémantiques différents : l'italien *carta* sera souvent associé à *papier* et à *carte*, amorçant ainsi la structuration de la signification en deux acceptations principales dont une désigne un ‘matériau’ l'autre un ‘support d'inscription’. Par suite, la confrontation avec l'anglais permet d'enrichir cette décomposition du sens : *carta* est souvent associé à *paper*, *card* ou *map*. Une troisième distinction apparaît, entre ‘document topographique’ et ‘petit support rectangulaire’ (correspondant aux cartes à jouer, au carte de visite, carte de crédit, etc.). On pourrait rétorquer que de telles relations ne nous permettent pas de distinguer entre la polysémie de *carta*, ou l'éventuelle synonymie de *paper*, *map* et *card*. Mais si l'on tient compte des correspondances de *paper*, *map* et *card*, dans d'autres langues, on obtiendra le plus souvent des équivalents différents (comme *papier*, *plan*, *carte*), ce qui permet d'affaiblir l'hypothèse de synonymie. Il est également possible de différencier polysémie et homonymie : dans la mesure où les liens polysémiques sont en partie motivés, ils sont fréquent de retrouver des polysémies parallèles (au moins partiellement) dans d'autres langues. Par exemple, les deux acceptations ‘document topographique’ et ‘petit support rectangulaire’ se retrouvent aussi bien dans le français *carte* que dans l'italien *carta*. Si ces deux sens correspondaient à des unités différentes homonymes, il serait étonnant que l'homonymie s'observe aussi bien en français qu'en italien, car l'homonymie est par définition fortuite (à la différence de la polysémie). Le schéma de la figure 1 montre comment le repérage de traduction permet de structurer les significations, à la manière de Hjelmslev (1971:113) lorsqu'il comparait la distribution de *bois* avec l'allemand *Holz* et *Wald*, et le danois *træ* et *skov*.

Italien	Français	Anglais
	papier	paper
carta	carte	map
		card

Figure 1. Des unités associables à *carta*.

Comme sur la figure 2, on peut observer des configurations complexes qui mettent en relation des niveaux distincts :

- entre les langues : on constate par exemple que *paper* partage de nombreuses acceptations avec *papier* (ce qui pourrait indiquer des significations voisines), malgré quelques différences ;
- entre chaque langue et les *designata* extra-linguistiques : cette relation, bien qu'invisible à l'intérieur des textes, peut être reconstruite grâce à certaines convergences (par exemple, lorsque *paper* est associé avec *article*, le *designatum* ‘article de presse’ ou ‘article scientifique’ peut être déduit sans équivoque) ;
- entre les unités d'une même langue : on constate la possible synonymie de *papier* avec *article*, mais aussi la divergence de leurs autres acceptations.

– entre les acceptations d'une même unité : la polysémie de *paper* ou de *papier* devient manifeste du fait de leur multiples possibilités de traduction.

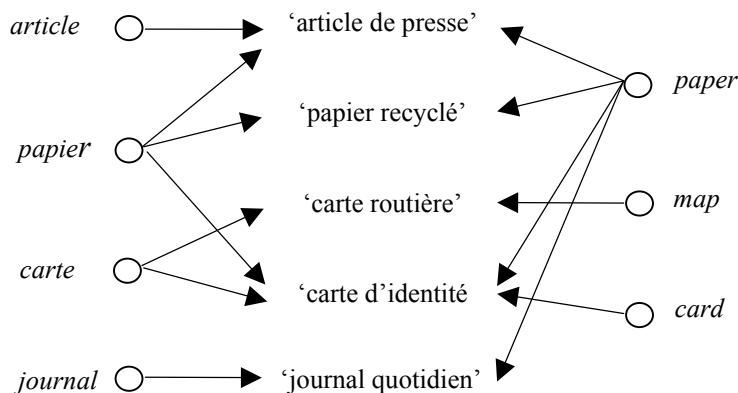


Figure 2. Réseau de relations interlingues manifestant des structures sémantiques.

Comme le note Pernier (1993 : 84), « la confrontation de signes appartenant à deux langues différentes révèle à la fois la polysémie de chacun (c'est-à-dire la diversité interne de leurs signifiés considérés du point de vue des concepts désignés) et la non-coïncidence de ces signifiés, c'est-à-dire le fait qu'ils sont polysémiques différemment. » La correspondance du français *disque* avec l'anglais *record* n'est valable que dans certains contextes, dans la mesure où les deux unités sont toutes deux polysémiques mais véhiculent des significations différentes : par exemple, en référence à un CD, *disque* indique la forme de l'objet, tandis que *record* s'attache à la fonction d'enregistrement. Chaque langue s'attache à des traits référentiels arbitrairement choisis, et le repérage de traduction permet d'objectiver les différences de choix. L'organisation particulière de chaque « système classificateur », selon l'expression de (Pernier 1993 : 109), devient ainsi manifeste : « Au niveau de son organisation interne, on pourrait dire que le signifié saisit les choses qu'il désigne non par leurs différences, mais par leurs ressemblances. Le passage de l'anglais au français n'a pas seulement pour effet de changer le signifiant ; il a pour effet de le faire changer de système classificateur. »

Ainsi, le repérage de traduction permet de comparer les codes, et de faire apparaître, pour chacun d'eux, des structurations sémantiques mises en lumière par la non-congruence des modes de désignation. Mais ce qui se dessine à travers cet entrelacs de liens interlinguistiques est extérieur aux codes eux-mêmes : ce sont les *designata* extra-linguistiques qui apparaissent en filigrane, puisqu'ils constituent le pivot de la relation d'équivalence traductionnelle. Considérons l'énoncé italien : *Questa carta è vecchia*. L'ensemble des *designata* potentiels de *carta* est très étendu : ‘papier’, ‘tapisserie’, ‘carte de crédit’, ‘carte à jouer’, ‘carte routière’, ‘carte de crédit’, ‘carte de visite’. Mais si l'on est en présence des traductions suivantes, l'ensemble des *designata* se réduit considérablement : *C'est une vieille carte / This is an old card / Esse bilhete e velho*. A l'intersection de toutes ces formulations linguistiques, toutes ambiguës si on les considère séparément, on trouve un *designatum* restreint, correspondant à ‘carte d’identité’. Comme le proposaient déjà Dagan et al. (1991), il est donc envisageable d’élaborer des méthodes de désambiguisation sémantique tirant parti des correspondances interlingues. La généralisation de ce principe à des corpus incluant plus de deux langues constitue un champ de recherche encore largement inexploré.

3 Le test de commutation interlingue

Le test de commutation interlingue a été suggéré par Catford (1965 : 28), afin de dégager des équivalences entre les unités d'un texte et de sa traduction : « Plutôt que de *se demander où* sont les équivalents, on peut adopter une procédure plus formelle, à savoir la commutation et l'observation de variations concomitantes. En d'autres termes, on peut introduire de manière systématique un changement dans le texte source et observer quels changements éventuels en découlent dans le texte cible. Un *équivalent de traduction textuelle* est donc : *cette portion du texte cible qui change si et seulement si une portion donnée du texte source a été modifiée.* » (nous traduisons) La même idée est à l'œuvre dans certaines méthodes d'alignement de textes parallèles, basées sur la reconnaissance des parties variables et les parties constantes dans un corpus d'exemples de traduction, afin d'établir des corrélations d'une langue à l'autre. Malavazos *et al.* (2000) en ont tiré une méthode d'extraction de « modèles de traduction » (*translation templates*) : « L'idée principale est basée sur le constat qu'étant donné une paire de phrases source et cible, toute modification de la phrase source aboutira probablement en un ou plusieurs changements dans la phrase cible, et qu'il est en outre probable que les unités constantes et variables de la phrase source correspondent respectivement aux unités constantes et variables de la phrase cible. » (nous traduisons). Des deux couples de phrases suivants, les auteurs tirent des correspondances entre les parties constantes et les parties variables :

angl.: Style Manager help menu
grec : Κατάλογος βοήθειας διαχειριτή ύφους

angl.: Style Manager file menu
grec : Κατάλογος αρχείων διαχειριτή ύφους

D'où les correspondances :

angl.: Style Manager X menu
grec : Κατάλογος X' διαχειριτή ύφους

(X,X') = (help, βοήθειας)
(X,X') = (file, αρχείων)

Dans ce derniers cas, les commutations ne sont pas produites, mais observées. De ce fait, elles peuvent être extraites automatiquement, par simple comparaison des phrases du corpus. Mais notons qu'il est peu probable qu'un corpus contienne en masse de tels cas de figure, où une seule unité est affectée par la commutation.

Le test manuel suit un parcours redoublé par rapport au test classique de commutation : en introduisant une variation sur le plan de l'expression on produit une variation sémantique ; cette variation sémantique impose ensuite une variation des signifiants cibles afin de rétablir l'équivalence sémantique entre les deux textes. Suivant ce principe, Mahimon (1999 : 37) propose une méthode dédiée à l'alignement manuel des unités lexicales, en reliant les unités qui commutent simultanément dans la source et la cible. Elle donne l'exemple suivant :

Fr.: Ce projet de loi prévoira un système de déclaration des maladies infectieuses
Angl.: This bill will provide for an infectious disease notification system

Si on fait commuter *Ce* avec *Chaque* l'équivalence peut-être rétablie en commutant *This* avec *Each* :

Fr.: **Chaque** projet de loi prévoira un système de déclaration des maladies infectieuses
Ang.: **Each** bill will provide for an infectious disease notification system

On peut en tirer des correspondances bilingues, que nous noterons de la manière suivante : *Ce* || *This*, *Chaque* || *Each*.

La commutation d'unités polylexicales s'effectue en plusieurs temps, par transitivité (si A et B commutent ensemble, et B et C commutent ensemble, alors A, B et C forment une unité).

Fr. : Ce projet de loi **prévoira** / **entérinera** un système de déclaration des maladies infectieuses
 Angl. :This bill will **provide for** / **confirm** an infectious disease notification system

d'où la commutation : *prévoira* || *provide for* (1)

Fr. : Ce projet de loi **prévoira** / **prévoit** un système de déclaration des maladies infectieuses
 Angl. :This bill **will provide** / **provides** for an infectious disease notification system

par conséquent, on a : *prévoira* || *will provide* (2)

Par transitivité *will provide for* est repéré comme une seule unité :

(1) + (2) \Rightarrow *prévoira* || *will provide for*

Sans rentrer dans les détails (voir Kraif 2001, pour une discussion de ce test), notons que ce test connaît des limites, surtout dans les cas de traductions « libres », où l'impossibilité d'établir des correspondances d'unité à unité le rend inapplicable. Mais la limitation la plus sévère reste son coût prohibitif, lorsqu'il est réalisé par des annotateurs humains.

4 Commutations interlingues distribuées

Il faut noter une extension très intéressante du test : l'extraction de correspondances lexicales. De nombreux travaux (Fung, 1994, Gaussier et Langé, 1995, Chang et Ker, 1996, McEnery et Oakes, 1996, Melamed, 1997a, Kraif 2001) ont montré qu'il est possible d'extraire des lexiques bilingues bruts à partir de l'observation des occurrences et des cooccurrences au sein d'un bi-texte. Toutes les méthodes ainsi développées se basent sur une idée simple : des unités source et cible qui apparaissent très fréquemment dans des segments équivalents (c'est-à-dire plus souvent que le hasard ne le laisserait escompter), sont vraisemblablement équivalentes.

(...u...,)
(...u..., ...u'...)
(... ...,)
(...u..., ...u'...)
(... ...,)
(... ..., ...u'...)
(...u...,)
(...u..., ...u'...)

Figure 3. Occurrences et cooccurrences de deux unités ($n_1=5$, $n_2=4$, $n_{12}=3$)

Dans l'exemple de la figure 3, on compte 5 occurrences de l'unité *u*, 4 cooccurrences de l'unité *u'* et 3 cooccurrences. En fonction des occurrences, on peut estimer le nombre de cooccurrences qu'on obtiendrait dans le cas d'une distribution aléatoire $(8*(5/8)*(4/8) = 2,5)$. Si le nombre de cooccurrences observées dépasse de manière significative cette estimation, on peut alors faire l'hypothèse que les unités sont des équivalents traductionnels. Plusieurs indices statistiques permettent de chiffrer la vraisemblance de cette hypothèse : l'information

mutuelle (Church 1990), le t-score (Fung et Church, 1994), le rapport de vraisemblance (Dunning, 1993) et la log-probabilité de l'hypothèse nulle (Kraif, 2001). Dans des travaux précédents (Kraif, 2001), nous avons mis en œuvre ces différents indices sur des unités lexicales manuellement tokenisées et lemmatisées. Les valeurs d'occurrences et de cooccurrences ont été calculées à partir des phrases du JOC (automatiquement alignées par nous). Pour chaque couple de phrases, nous avons appliqué l'algorithme de meilleure affectation biunivoque (noté ABIJ) : 1/ calcul de l'indice d'association pour tous les appariements possibles d'unités ; 2/ sélection et enregistrement du couple d'unités obtenant le meilleur indice ; 3/ élimination, dans l'ensemble des couples candidats, de tous les couples concurrents du couple sélectionné (i. e. qui mettent en jeu une des deux unités sélectionnées) ; 4/ tant qu'il reste des candidats, retour en 2. Les résultats obtenus ont été évalués sur un corpus de référence d'environ 700 couples de phrases alignés manuellement (aléatoirement tirées du corpus JOC). Les jeux d'appariements obtenus automatiquement ont été comparés avec les couples de référence, en calculant la précision P (nombre de couples corrects/nombre de couples extraits), le rappel R (nombre de couples corrects/nombre de couples de référence) et la F-mesure pour synthétiser P et R (la moyenne harmonique égale à $2PR/(P+R)$).

Les valeurs de F-mesure des extractions réalisées avec ces différents indices sont représentées figure 4.

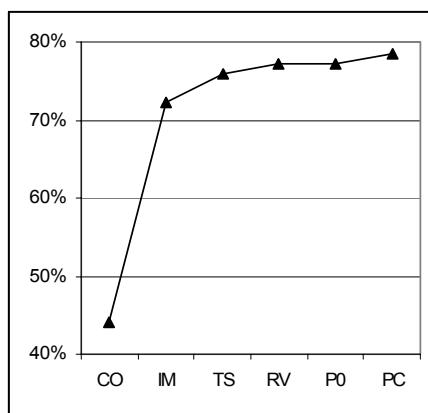


Figure 4. F-mesure des extractions de correspondances lexicales. CO : indice basé sur la cognition (mots apparentés), IM : information mutuelle, TS : T-score, RV : rapport de vraisemblance, P0 : log-probabilité de l'hypothèse nulle, et PC : combinaison de CO et P0.

Avec PC, indice basé sur les cooccurrences et l'identification des cognats, nous avons obtenu des résultats très satisfaisants ($F = 78,5\%$). Notons que la seule observation des cooccurrences permet d'obtenir presque aussi bien ($F = 77,2\%$). Or, il apparaît que ce type d'observation n'est rien d'autre qu'une extension du test de commutation, mais en négatif : on s'appuie sur le nombre de fois que les contextes des unités commutent, quand les unités apparaissent ensemble, rapporté au nombre de fois où, dans des contextes équivalents, les unités apparaissent séparément. Comme dans le test de commutation classique, ce sont les variations concomitantes qui permettent de dessiner l'organisation des unités à travers le jeu des identités et des différences. Mais cette fois-ci, ces variations ne sont plus appréhendées dans le cadre d'une série d'observations discrètes. Le filtrage de la masse des occurrences et des cooccurrences révèle des *régularités* et non des *règles*. Les corrélations étudiées entre les deux plans parallèles, c'est-à-dire les deux idiomes confrontés dans la relation de traduction, ne relève pas d'une loi du tout ou rien, comme dans la commutation de *vache* avec *bâche*. Ce qui nous intéresse dans cette masse de commutation, c'est qu'elle prend une forme à mesure

qu'elle croît, qu'elle exhibe des régularités qui ne peuvent être imputables au hasard, ni aux choix individuels du traducteurs, ni aux contingences de la situation de communication. Au dessus du « bruit » traductionnel, ces régularités révèlent les points de contact entre les codes : elles filtrent finalement ce qui dans la traduction ressort au transcodage. Il existe une manière objective de quantifier cette propriété des corpus de traduction. Si l'on compare un jeu de correspondances lexicales manuellement extraites avec un jeu d'appariements tirés au hasard à l'intérieur de phrases alignées, une différence formelle apparaît immédiatement : les couples corrects présentent beaucoup plus de répétitions, d'« ordre », que les couples pris aléatoirement. Par exemple, si l'on examine les 10 occurrences de *against* dans notre corpus de référence, on dénombre seulement 3 paires différentes, tandis qu'avec un tirage aléatoire des appariements on en a obtenu 10, comme le montre le tableau de la figure 5.

<i>Correspondances extraites manuellement</i>	<i>Correspondances extraites aléatoirement</i>
(against, à l'encontre de)	(against, par)
(against, à l'encontre de)	(against, procédure)
(against, à l'encontre de)	(against, moratoire)
(against, au détriment de)	(against, à l'encontre de)
(against, contre)	(against, dont)
(against, contre)	(against, contre)
(against, contre)	(against, effectivement)
(against, contre)	(against, charges)
(against, contre)	(against, Etat membre)
(against, contre)	(against, qui)

Figure 5. Correspondances lexicales correctes vs aléatoires

Pour quantifier ce type de dispersion, nous proposons de calculer l'entropie conditionnelle, qui mesure le « désordre » des cooccurrences de deux unités source et cible, par rapport aux occurrences de l'une ou de l'autre. Les équations [1] et [2] donnent l'expression de l'entropie conditionnelle dans les deux sens de la traduction :

$$H(T'/T) = -\sum_u p(u) \sum_{u'} p(u'/u) \log p(u'/u) = -\sum_u \sum_{u'} p(u, u') \log \frac{p(u, u')}{p(u)} \quad [1]$$

$$H(T/T') = -\sum_{u'} p(u') \sum_u p(u/u') \log p(u/u') = -\sum_{u'} \sum_u p(u, u') \log \frac{p(u, u')}{p(u')} \quad [2]$$

où T et T' sont respectivement les textes source et cible, u et u' des unités de T et T' , $p(u)$ représente la probabilité d'apparition de u à gauche d'un couple d'unités appariées, $p(u')$ la probabilité d'apparition de u' à droite d'un couple d'unités appariées, et $p(u, u')$ la probabilité de l'appariement (u, u') . Afin d'étudier la corrélation entre cette quantité et la correction des résultats, nous avons évalué les valeurs d'entropie pour différentes séries d'extractions de correspondances comportant différentes proportions d'erreurs. On a ainsi obtenu 6 séries d'extractions (pour plus de détail, cf. Kraif 2001) :

- Les appariements de référence extraits manuellement ;

- 6 extractions (pour les indices CO, TS, IM, RV, P0, PC), avec l'algorithme d'association maximale AMAX² ;
- 6 extractions (pour les indices CO, TS, IM, RV, P0, PC), avec l'algorithme ABIJ, (analogue à celui décrit par Melamed, 1997a) ;
- 7 extractions obtenues avec 7 pondérations différentes d'un indice combinant P0 à une valeur aléatoire³.
- 6 extractions filtrées⁴ (pour les indices CO, TS, IM, RV, P0, PC), avec AMAX ;
- 6 extractions filtrées (pour les indices CO, TS, IM, RV, P0, PC), avec ABIJ.

La figure 6 montre une étroite corrélation entre l'entropie conditionnelle⁵ et la précision de chaque jeu de correspondances. Le coefficient de corrélation linéaire est en effet de -0,96. Malgré les choix de traduction particuliers, il existe bien des régularités quantifiables. La variabilité traductionnelle constitue un « bruit » au dessus duquel émergent les structurations des codes. Au delà du niveau de la *parole*, se dessinent les contraintes des *langues*. Au delà des effet de *sens*, s'affirment les *significations*.

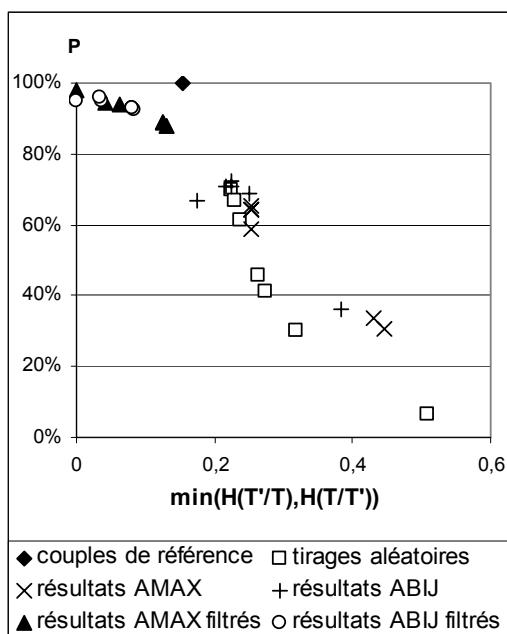


Figure 6. Corrélation entre la précision des extractions et leur entropie conditionnelle

Le repérage de traduction comporte donc deux faces : une face subjective, en tant qu'il nécessite l'interprétation d'un sujet pour relier des unités équivalentes d'un point de vue traductionnel dans un contexte particulier ; et une face objective, en tant que certaines

² A la différence de ABIJ, avec AMAX, pour chaque unité source, on sélectionne l'appariement qui a obtenu la meilleure valeur de l'indice. Une même unité cible peut donc apparaître dans plusieurs couples. Ainsi AMAX est dissymétrique vis-à-vis des deux textes.

³ Soit l'indice AL = (1-Coeff)*P0 + Coeff*Random, où Random est une valeur aléatoire comprise entre 0 et 10, et Coeff prend les valeurs respectives : 0.25, 0.5, 0.75, 0.95, 0.97, 0.99, 1

⁴ Les extractions filtrées ne retiennent que les couples ayant obtenu un indice au moins deux fois supérieur à tous leurs concurrents. Elles présentent en général une précision supérieure pour un rappel dégradé.

⁵ Pour chaque extraction, nous avons pris $\min(H(T/T'), H(T'/T))$, c'est-à-dire qu'à chaque fois, nous avons favorisé le sens de traduction où les régularités apparaissaient avec plus de force.

correspondances manifestent des régularités (correspondant à un minimum d'entropie) que l'on peut extraire automatiquement avec des méthodes fiables.

5 Conclusion

Le repérage de traduction apparaît comme une tâche incontournable dans l'exploration des corpus multilingues alignés. Sa mise en oeuvre aboutit à la définition d'unités polylexicales pertinentes comme unités de traduction, donnant des indices intéressants sur des composés de nature phraséologique ou terminologique. Dans la mesure où il débouche sur l'extraction de séries d'équivalences, il fournit un matériau empirique de premier choix pour l'aide à la traduction, la lexicographie bilingue ou la terminologie. En outre, par l'étude distributionnelle des correspondances, il permet d'étudier des organisations sémantiques complexes à travers le prisme des contrastes interlingues. Enfin, la traduction aboutit à une certaine explicitation du sens : le repérage de traduction peut conduire alors à une désambiguïsation partielle des unités.

Dans la tentative de systématiser le repérage de traduction, nous avons vu que le test de commutation interlingue fournissait des critères intéressants pour l'observation manuelle. Mais ce test est difficilement automatisable, car il fait intervenir implicitement les compétences du locuteur pour le choix des unités et le rétablissement des équivalences. Et il est rare qu'un corpus soit assez riche et vaste pour contenir un exemple déjà existant permettant la mise en oeuvre du test. Nous montrons cependant que les techniques d'extraction de correspondances lexicales, basées sur des observations distributionnelles, constituent un prolongement possible du test, et permettent d'obtenir un repérage fiable pour une bonne partie des unités. En outre, le recours aux méthodes d'extraction automatique présente un avantage : il permet de ne conserver que les appariements les plus significatifs (au sens statistique), et d'éliminer les traductions dues à des choix locaux difficiles à interpréter en dehors d'un contexte particulier.

L'observation de ces régularités est selon nous encore largement sous-exploitée : en appariant des unités comportant des annotations morphologiques, grammaticales ou sémantiques, le repérage de traduction fournit un outil permettant d'étudier une vaste gamme de phénomènes contrastifs (concordances des temps, diathèse, etc.). A travers des recherches en cours, telles que les projets Arcade 2 ou Carmel⁶, nous espérons identifier plus précisément quelles sont les formes de régularités que les méthodes quantitatives peuvent saisir, et quelle interprétation en donner.

Références

Catford J. C. (1965) *A Linguistic Theory of Translation*, London, Oxford University Press.

Chang J. J. S., Ker S. J. (1996) Aligning More Words with High Precision for Small Bilingual Corpora, *Proceedings of the 16th International Conference on Computational Linguistics, COLING-96*, Copenhagen, 5-9 August 1996.

⁶ Ces deux projets, respectivement coordonnés par le DELIC et le LIA, sont actuellement dans leur phase initiale et s'inscrivent dans le cadre de l'appel d'offre Technolangue.

Church K. W., Hanks P.(1990) Word Association Norms, Mutual Information, and Lexicography, *Machine Translation*, vol. 16, n. 1, p. 22-29.

Dagan I., Itai A., Shwall U. (1991) Two Languages Are More Informative Than One. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, ACL-91*, Morristown, NJ, pp. 130-137.

Dunning T. (1993) Accurate Methods for the Statistics of Surprise and Coincidence, *Computational Linguistics*, vol. 19, n. 1, Morristown, NJ, p. 61-74.

Fung P., Church K.W. (1994) K-vec : A New Approach for Aligning Parallel Texts, *Proceedings of the 15th International Conference on Computational Linguistics*, p. 1096-1102.

Gaussier E., Langé J.-M. (1995), Modèles statistiques pour l'extraction de lexiques bilingues, *T.A.L.*, Vol. 36, N° 1-2, p. 133-155.

Greimas A. J. (1970) *Du Sens, Essais sémiotiques*, Paris, Editions du Seuil.

Hjelmslev L. (1971) *Essais linguistiques*, Paris, Editions de Minuit.

Isabelle P. (1992) La bi-textualité : vers une nouvelle génération d'aides à la traduction et la terminologie, *META*, Vol. XXXVII, N°4, Outremont, Canada, p. 721-731.

Kraif O. (2001) *Constitution et exploitation de bi-texte pour l'aide à la traduction*, Thèse de doctorat, Université de Nice. URL : <http://www.u-grenoble3.fr/kraif>.

McEnery A. M., Oakes M. P. (1996) Sentence and word alignment in the CRATER project, In Thomas J., Short M. (Ed.), *Using Corpora for Language Research*, London, Longman.

Mahimon, M.-D. (1999) *Identification des équivalences traductionnelles sur un corpus Français / Anglais*, Mémoire de DEA sous la dir. de Jean Véronis, Université de Provence

Malavazos C., Piperidis S., Carayannis G. (2000) Towards memory and template-based translation synthesis, *Proceedings of MT 2000*, 20-22 November 2000, Exeter, UK.

Melamed I. D. (1997a) A Word-to-Word Model of Translational Equivalence, *35th Conference of the Association for Computational Linguistics (ACL'97)*, Madrid.

Melamed, I. D. (1997b) Automatic Discovery of Non-Compositional Compounds in Parallel Data. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, 1-2 August 1997, pp. 97-108

Pergnier M. (1993) *Les fondements sociolinguistiques de la traduction*, Lille, Presses Universitaires de Lille.

Vinay J.-P, Darbelnet J. (1959) *Stylistique comparée du français et de l'anglais*, Paris, Didier.

Un outil d'extraction terminologique endogène et multilingue

Jacques Vergne

GREYC - UMR 6072

campus II - BP 5186

Université de Caen

F-14032 CAEN cedex

FRANCE

www.info.unicaen.fr/~jvergne

e-mail : Jacques.Vergne@info.unicaen.fr

tél. : 02 31 56 73 36

Résumé – Abstract

Dans cet article, nous présentons un outil d'extraction terminologique "endogène" à partir d'un corpus multilingue. Cet outil est qualifié d'endogène car, sans autre ressource que le corpus dont il doit extraire les termes, il calcule les mots vides à partir de ce corpus pour centrer les termes candidats sur des mots pleins. Il est placé dans le cadre d'un système de constitution automatique de revue de presse à partir de sites de presse présents sur l'internet¹. Il s'agit de répondre à des questions telles que : "de qui, de quoi est-il question aujourd'hui dans la presse de tel espace géographique ou linguistique ?". Le corpus est constitué des textes des hyperliens des "Unes" des sites de presse de langues inconnues a priori. Il est renouvelé quotidiennement, et sa taille est d'environ 100 Ko (débalisé). La méthode est fondée sur l'analyse distributionnelle, et utilise des différences entre mots contigus : les différences de longueur et d'effectif.

In this paper, we present an "endogenous" terminology mining tool, from a multilingual corpus. This tool is described as endogenous because, without any other resource than the

¹ Une démonstration est accessible sur :

www.info.unicaen.fr/~jvergne/demoRevueDePresse/index.html

corpus from which it has to extract terms, it computes function words from this corpus to focus candidate terms on content terms. It is used inside an automatic news review system from news web sites. The system is able to answer questions as : "who, what are newspapers speaking about today in a given geographic or linguistic search space?". The corpus is made of hyperlinks texts of news web site front-pages in unknown languages. It is daily downloaded, and its size is about 100 Kbytes (untagged). The method is based on distributional analysis, and uses differences between contiguous words : differences of length and of frequency.

Mots Clés – Keywords

extraction terminologique, endogène, multilingue, internet, fouille de texte.

terminology mining, endogenous, multilingual, internet, web mining, text mining.

1 Introduction

Le présent travail s'inscrit à l'intersection des TALN et du "web mining" car il applique des concepts issus de l'informatique linguistique à la problématique du web mining. Ce domaine de recherches mixte se développe actuellement, à l'exemple du groupe CLAIR : Computational Linguistics And Information Retrieval group at the University of Michigan (perun.si.umich.edu/clair/) dont les thèmes de recherches principaux concernent les systèmes questions - réponses et le résumé automatique. Cette problématique impose des contraintes qui lui sont propres : multilinguisme généralisé, production quotidienne et massive d'informations surtout textuelles, et besoins nouveaux des utilisateurs d'accès toujours plus efficace et plus précis à ces informations. Cette problématique nous donne la possibilité de concevoir de nouvelles tâches qui constituent des enjeux très intéressants pour les TALN. Nous allons d'abord présenter l'application cadre (section 2), puis les spécifications de l'outil (section 3). Puis nous proposons une manière originale de poser le problème comme un calcul de différences entre mots (section 4). Ensuite une solution est proposée et décrite (section 5). Nous présentons enfin des résultats et leur évaluation (section 6) puis une discussion sur la méthode (section 7).

2 L'application cadre

L'application cadre est un système de constitution automatique de revue de presse à partir de sites de presse présents sur l'internet, pour des utilisateurs qui se demandent de qui et de quoi il est question aujourd'hui dans la presse de tel espace géographique ou linguistique. Ce système inverse la problématique des moteurs de recherche : au lieu de rechercher des

documents à partir de mots-clés qui représentent des thèmes, il s'agit de produire **en sortie** les thèmes principaux de l'actualité, et de donner accès aux articles concernés par ces thèmes (à la manière de Google News : news.google.fr). Pour chaque site, **un seul document** est téléchargé : le document du point d'entrée de chaque site de presse, c'est-à-dire sa "Une". De ce document, sont extraits les hyperliens : les URL et le code source des "textes" de liens. Ces codes source de "texte" de liens sont composés de titres ou de résumés d'articles (avec leur mise en forme), et d'URL vers des images, des photographies ou des icônes. C'est de ces textes de liens (leur code source débalisé) que sont extraits les termes candidats. Ne sont retenus comme termes que les termes candidats **présents sur plusieurs sites**. Les URL des articles ne servent qu'en sortie, pour donner accès à un article, si l'utilisateur le décide. Le système ne se sert pas des articles eux-mêmes. Cette économie de traitement s'appuie sur le fait que la rédaction d'un texte de lien est un choix éditorial des journalistes des sites de presse. Le système calcule un graphe de termes dans lequel les nœuds sont les termes et les arcs sont les relations entre termes, relations définies par la co-occurrence de deux termes dans un même texte de lien. L'utilisateur peut naviguer dans ce graphe pour accéder à des termes liés et à des articles (à la manière de Kartoo : www.kartoo.com).

3 Spécifications de l'outil

Les spécifications de l'outil viennent de l'application cadre : il s'agit d'extraire des termes de petits corpus multilingues : les **corpus des textes d'hyperliens** collectés quotidiennement sur les Unes de sites de presse. Ces corpus sont thématiquement variés, et relativement petits (environ 80 à 170 Ko, 15000 à 30000 mots). Les langues sont alphabétiques, inconnues a priori, mélangées dans le corpus, et non diagnostiquées dans les calculs. Les calculs sont indépendants des langues, et donc insensibles à l'ajout d'une nouvelle langue et aux proportions entre langues différentes. Les tests ont été faits sur des corpus comprenant surtout du français, de l'anglais, de l'allemand, de l'italien, et de l'espagnol (un nouveau corpus acquis chaque jour). La tâche revient à distinguer les mots vides des mots pleins², pour centrer la construction des termes candidats sur les mots pleins. La méthode ne doit pas utiliser de ressources propres à une langue, pour ne pas avoir à faire un travail de préparation de ressources linguistiques à chaque nouvelle langue traitée (ouverture dans chaque langue, et ouverture à d'autres langues).

² Nous avons choisi dans cet article les termes : "mot vide" - "mot plein", synonymes de "mot grammatical" - "mot lexical" et de "function word" - "content word", à la suite de Lucien Tesnière et de Fathi Debili, et aussi dans la tradition de l'informatique documentaire, où un "mot vide" est un mot qui ne doit pas être indexé, qu'il soit mot grammatical ou mot lexical non discriminant. Les mots vides sont alors souvent regroupés dans une "stoplist" ou "ante-dictionnaire".

4 Comment poser le problème

Plusieurs méthodes ont été explorées, en nous imposant la contrainte de trouver une méthode n'utilisant aucune autre ressource linguistique que le corpus traité lui-même (méthode appelée pour cela "endogène") : la recherche des motifs répétés par l'algorithme glouton (recherche des n-grammes à partir des n-1-grammes) a été expérimentée, en excluant les mots vides par leur fréquence (test de Zipf). Les résultats étaient corrects, mais le départage entre les mots vides rares et les mots pleins très fréquents n'était pas possible (tels que *guerre*, *war*). La fréquence des mots n'est pas un indice suffisant : il faut s'intéresser aux fréquences des mots **à certaines positions**. D'où une autre manière de poser le problème : comment distinguer les mots pleins des mots vides à partir du corpus ? Notre direction de travail est d'utiliser à la fois les formes et leurs positions, ou plus abstrairement le concept de différence (ou de valeur relative), une constante du Thème Syntaxe et Rhétorique du GREYC, dont Hervé Déjean (Déjean, 1998), Nadine Lucas (Lucas, 2001), alors que le test de Zipf n'utilise que les fréquences des formes, sans exploiter leurs positions relatives. Or une observation fondamentale de Zipf lui-même est que les mots vides sont fréquents et courts et que les mots pleins sont plus rares et plus longs (ce qui est d'usage fréquent est court : c'est la loi de l'économie d'effort dans l'usage d'un code, caractérisée par Zipf (Zipf, 1949), et observable aussi dans les langages de programmation). D'où l'idée d'utiliser les différences entre mots³ : non plus seulement les différences de fréquence, mais aussi les différences de longueur. Nous allons donc considérer le texte comme une suite de mots vides et de mots pleins, plus précisément comme une suite de mots vides et de mots non vides, car tout mot non vide est considéré comme plein. Voici un segment de texte (La Stampa du 15 mars 2003) :

Manifestazioni per la pace in tutto il mondo

L'outil donne **en sortie** le résultat suivant :



Manifestazioni per la pace in tutto il mondo

où chaque mot est symbolisé par un ovale blanc pour les mots vides, et un ovale noir pour les mots non vides. Notre problème devient alors un problème de détection de frontière. Les frontières sont caractérisées par une **différence orientée** sur l'axe syntagmatique : noir - blanc, ou blanc - noir. On en vient donc plus précisément à une détection de différence(s). Voici les critères de différence entre 2 mots contigus :

³ En application du principe bien connu de Saussure : "dans la langue il n'y a que des différences." (Saussure, CLG, p.166).

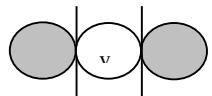
- différence de longueur en nombre de lettres : ***Manifestazioni per*** (14 lettres - 3 lettres)
- différence d'effectif dans le corpus : ***il mondo*** (19 occurrences - 3 occurrences)

Nous allons utiliser plusieurs différences sur plusieurs mots contigus, ce qui revient à étudier conjointement les dérivées des fonctions "longueur des mots" et "effectif des mots" selon leur position sur l'axe syntagmatique.

5 Solution proposée

5.1 Principes et propriétés

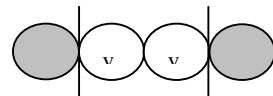
Sans autre donnée que les graphies elles-mêmes, nous allons détecter deux types de séquences de mots, où un ou deux mots vides sont encadrés par deux mot non vides (d'où un couple de 2 frontières), ce que nous illustrons par deux exemples de séquences à détecter :



tutto il mondo

profil des longueurs : long - court - long

profil des effectifs : rare - fréquent - rare



Manifestazioni per la pace

long - court - court - long

rare - fréquent - fréquent - rare

On détecte la séquence PVP par son profil caractéristique long - court - long et rare - fréquent - rare sur les fonctions longueur et effectif, et la séquence PvVP par son profil caractéristique long - court - court - long et rare - fréquent - fréquent - rare. Les deux types de séquence sont détectés sur des profils convergents des deux fonctions longueur et effectif (& booléen sur les deux critères). L'utilisation de deux critères améliore la robustesse des calculs, mais chacun pris séparément donne déjà de bons résultats. Le plus souvent, les résultats selon les deux critères sont convergents, ce qui corrobore la propriété énoncée par Zipf.

5.2 Processus et algorithme

Le processus comporte 2 étapes : étudier le corpus pour en extraire les mots vides, et générer les termes candidats.

5.2.1 Étudier le corpus pour en extraire les mots vides

Le corpus est segmenté sur les limites de textes de liens et sur les ponctuations, d'où des segments physiques délimités des ponctuations, que nous nommons "virgulots". Pour chaque virgulot, on passe les 2 filtres selon les 2 séquences.

Pour détecter une séquence PvP, on vérifie les profils long - court - long **et rare - fréquent - rare**, c'est-à-dire que la **longueur** du mot central est **inférieure** aux longueurs des 2 mots qui l'entourent, **et** que son **effectif** est **supérieur** à leurs effectifs. Pour détecter une séquence PvVP, on vérifie de la même manière les profils long - court - court - long **et rare - fréquent - fréquent - rare**. Voici un exemple de détection des deux séquences dans le même virgulot :

	<i>Manifestazioni</i>	<i>per</i>	<i>la</i>	<i>pace</i>	<i>in</i>	<i>tutto</i>	<i>il</i>	<i>mondo</i>
longueurs	14	3	2	4	2	5	2	5
profils	long	court	court	long		long	court	long
effectifs	1	10	207	2	62	3	19	3
profils	rare	fréquent	fréquent	rare		rare	fréquent	rare
déductions		mot vide	mot vide				mot vide	

Les mots sont catégorisés mots vides sous les deux formes graphiques : tout en minuscules, et avec une initiale majuscule. On trouve en section 6 les mots vides les plus fréquents extraits le 15 mars 2003 des deux espaces de recherche servant aux tests.

5.2.2 Générer les termes candidats

Les mots non vides sont considérés comme des mots pleins et peuvent devenir terme candidat. Les termes candidats sont extraits de chaque virgulot, dont les mots sont étiquetés vides ou non vides, pour générer des termes selon les motifs suivants :

P+	<i>Manifestazioni</i>	<i>pace</i>	<i>tutto</i>	<i>mondo</i>
P+ v+P+	<i>Manifestazioni per la pace</i>	<i>pace in tutto</i>	<i>tutto il mondo</i>	
P+ v+P+ v+P+	<i>Manifestazioni per la pace in tutto</i>	<i>pace in tutto il mondo</i>		

6 Résultats et évaluation

Voici les résultats des traitements des corpus téléchargés le 15 mars 2003 à partir des deux espaces de recherche servant aux tests, avec les termes candidats les plus fréquents (avant la suppression des termes présents sur un seul site) :

espace de recherche	espace de recherche 1 22 sites de la presse française nationale et régionale, 17 sites de la presse européenne (Suisse, Belgique, Allemagne, Italie, Espagne, UK, Irlande), et 4 sites de presse nord-américaine, chaque langue étant représentée par au moins deux sites	espace de recherche 2 une centaine de sites publiés par Google News, environ la moitié étant des sites nord-américains, le reste du monde entier (news.google.fr/news/)

corpus	84 Ko, 14 800 mots				163 Ko, 28 500 mots			
termes candidats	1566 occurrences de 584 termes candidats (de 42 à 2 occ. / terme)				2435 occurrences de 820 termes candidats (de 47 à 2 occ. / terme)			
termes candidats les plus fréquents	article : 42 guerre : 21 Jean-Luc Lagardère : 17 monde : 12 Açores : 11	Weitere Artikel : 10 mort : 10 Bagdad : 8 empire : 8 semaine : 8 Lettre : 7	Plan : 7 fin : 7 guerra : 7 procès : 7 réforme : 7 sommet : 7 Echos : 6	Läs mer: 47 ÉÑ Ä : 29 Laden : 24 war : 22 Kabul : 20 Qaeda : 20 China : 18	Statement : 17 Sep 12 : 15 Pak : 14 Press Secretary : 13	Sep 11 : 13 Northern Alliance: 12 guerra : 12 Irak : 11 Kandahar : 11		
mots vides les plus fréquents	de : 340 la : 207 l' : 153 le : 113 d' : 107 à : 107	du : 103 et : 99 des : 88 en : 87 les : 84 a : 82	un : 80 Le : 74 La : 72 L' : 62 in : 62 une : 56	Les : 55 's : 55 to : 53 pour: 43 au : 41 sur : 41	to : 327 in : 280 of : 237 the : 230 's : 166 de : 154	for : 144 on : 143 and: 138 a : 126 The: 118 en : 76	la : 75 by : 55 Al : 53 with: 52 is : 41 A : 38	from: 36 at : 34 i : 34 't : 32 un : 31 à : 31
termes candidats extraits à tort : bruit causé par un silence de la détection des mots vides	Was : 5 Tutti : 4 vous : 3 About: 2 Alors : 2 Ein : 2 Have : 2	If : 2 Mais : 2 Qu' : 2 Wie : 2 Wo : 2 avant : 2 contra: 2	could : 2 depuis:2 encore:2 faut : 2 mieux: 2 nous : 2 now : 2	plusieurs : 2 that : 2 tout : 2 tutto : 2	This: 12 How : 7 Don' : 6 It : 6 Most : 4 contra: 4	won' : 4 Alla : 3 My : 3 auf : 3 One : 2 Wer : 2	Where:2 Why : 2 après : 2 down : 2 einer : 2	enough: 2 only : 2 they : 2 when : 2 which: 2
	25/584 = 4,3% des 584 termes candidats extraits				22/820 = 2,7% des 820 termes candidats extraits (résultats sous-évalués à cause de quelques langues inconnues)			
termes candidats non extraits : silence causé par un bruit de la détection des mots vides	War : 9 paix : 7 soir : 7 war : 7 aide : 4	dimanche: 4 Photo : 3 baisse : 3 Aide : 2 Groupe : 2	attendu : 2 home : 2 turn : 2 voie : 2 world : 2	News : 77 New: 43 news : 23 killed : 18 Home : 17	Help : 16 Free : 10 Global : 9 Air : 8 help : 8	make : 8 First : 7 Get : 7 get : 7 groups : 7 ...		
	15/584 = 2,6% des 584 termes candidats extraits				88/820 = 10,7% des 820 termes candidats extraits			
termes retenus les plus fréquents (nb de sites - nb d'articles)	guerre (12-24) Lagardère (11-16) Jean-Luc Lagardère (9-12) monde (8-13) 15 (7-10) 16 (7-9) Aznar (7-8) Açores (7-10) empire (7-8)	semaine (7-8) Chirac (6-6) Premier ministre (6-7) fin (6-9) français (6-9) mort (6-10) pays (6-10) site (6-8) sommet (6-6) ...	Policy (19-23) U.S. (18-39) China (14-29) war (14-71) Special (12-24) This (12-24) United (12-18) Privacy Policy (11-11) Week (11-14)	East (10-12) American (9-14) Information (9-13) Press (9-25) Saddam (9-13) Azores (8-8) How (8-10) Index (8-8) Middle East (8-8) Money (8-8) ...				

N.B. au sujet de l'évaluation : comme les termes candidats sont construits sur les mots non vides, silence et bruit sur la détection des mots vides entraînent respectivement bruit et silence sur la génération des termes candidats.

Revenons à un de nos objectifs de départ : les mots vides rares et les mots pleins très fréquents sont-ils correctement repérés ? La méthode proposée, parce qu'elle utilise un calcul fondé sur des différences entre mots et non des valeurs absolues, rend la détection des mots vides presque indépendante de leur effectif. Par exemple, les mots *article* (42 occurrences), *guerre* (21), *monde* (12), *mort* (10), *guerra* (7) sont très fréquents et correctement détectés mots pleins, alors que *della* (6), *sous* (5), *bei* (4), *our* (3), *eines* (2), *Vers* (1) sont peu fréquents et correctement détectés mots vides : il suffit qu'un seul contexte ait pu les détecter au moyen des différences adéquates.

7 Discussion

L'idée d'utiliser le matériau linguistique traité lui-même pour en extraire des ressources nécessaires à ce traitement est déjà ancienne : on la trouvait dès 1982 chez Fathi Debili (Debili, 1982), puis chez Didier Bourigault dans LEXTER (Bourigault, 1994, pp. 63-78) et dans SYNTTEX (Bourigault, 2002), et aussi chez François Rousselot (Frath, Oueslati, Rousselot, 2000). Cependant, il faut préciser quelles ressources sont extraites, et pour réaliser quel traitement : Bourigault et Rousselot utilisent des régularités distributionnelles lexicales pour extraire des termes candidats d'un corpus, et calculer les rattachements des groupes prépositionnels, à partir d'un vaste corpus monolingue et très cohérent thématiquement. Dans le travail ici présenté, nous utilisons aussi des régularités distributionnelles, mais la tâche est différente : il s'agit plus simplement de discriminer les mots vides pour construire les termes candidats à partir des mots pleins. Les prétraitements sont aussi différents : Debili et Bourigault font d'abord une analyse syntaxique, Rousselot et Helena Ahonen (Ahonen-Myka, 1999) recherchent les motifs répétés (algorithmes extrapolés de l'algorithme glouton) et mettent en entrée les mots vides pour éviter de les prendre comme termes (stopword-list), ce qui dans les deux cas nécessite la connaissance de la langue unique traitée, et la constitution manuelle des ressources propres à cette langue. Mais l'outil présenté ne nécessite **aucun prétraitement ni aucune ressource préalable**. Il traite un **corpus multilingue**, c'est-à-dire où les langues sont mélangées, inconnues, et qui ne sont à aucun moment diagnostiquées. Cet outil se situe dans la lignée des travaux d'Hervé Déjean (Déjean, 1998), qui a proposé une méthode de "découverte des structures formelles des langues", sur corpus bruts monolingues, de langues très variées, sans prétraitement ni ressources préalables.

Nous avons ainsi repris le terme "**endogène**" proposé par Didier Bourigault, dans le même sens générique, mais dans un sens spécifique différent.

8 Conclusion et perspectives

Nous avons présenté un outil d'extraction terminologique utilisant une méthode générique sur la dimension des langues, n'utilisant pas d'analyse syntaxique, ni de dictionnaire, ni de stoplist et capable de repérer à la fois les mots vides rares et les mots pleins très fréquents, dans un corpus multilingue, de langues alphabétiques, inconnues a priori, mélangées dans le corpus, et non diagnostiquées dans les calculs. Les calculs sont indépendants des langues, et ne sont donc sensibles ni à l'ajout d'une nouvelle langue, ni aux proportions entre langues différentes. La bonne qualité des résultats et l'adéquation de la méthode à la tâche nous montre que des propriétés linguistiques très générales sont exploitées : principalement les différences (ou valeurs relatives). Notons qu'on ne s'est pas intéressé à la distinction nominal - verbal, cette distinction étant inutile dans la tâche. Le fait de ne pas s'occuper de la distinction verbo-nominale ne perturbe pas l'exécution de la tâche, car on ne retient que les termes présents au moins deux fois dans le corpus (ensuite sur deux sites différents); d'autre part les groupes nominaux sont fréquents et répétés, les groupes verbaux sont plus rares et plus variés, d'où très peu de groupes verbaux en sortie.

De nombreuses directions restent à explorer. Au sujet du calcul des différences entre mots, peut-on définir d'autres critères ? Pour mieux traiter les mots vides rares, nous envisageons deux passes : une pour les mots pour lesquels il n'y a pas eu de contradiction entre critères de différence (traitement actuel), la deuxième les utilisant pour catégoriser les autres mots; une solution complémentaire serait de mémoriser les déductions faites la veille. Enfin, la méthode a été confrontée à un but opératoire, mais elle pourrait être approfondie en tant qu'exploration linguistique indépendante des langues, par exemple en direction de la catégorisation verbo-nominale.

Références

Ahonen-Myka H., Heinonen O., Klemettinen M., Verkamo A. I. (1999), Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery, Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining : Foundations, Techniques and Applications, ed. R. Feldman, 1-9.

www.cs.helsinki.fi/u/hahonen/ham_ijcai99.ps

Bourigault, D. (1994), *Lexter, un logiciel d'extraction de terminologie. Application à l'acquisition des connaissances à partir de textes*, Thèse de l'Ecole des Hautes Etudes en Sciences Sociales, Paris.

Un outil d'extraction terminologique endogène et multilingue

Bourigault, D. (2002), Upéry : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus, Actes de la 9ème conférence annuelle sur le Traitement Automatique des Langues (TALN 2002), Nancy, 75-84.

www.univ-tlse2.fr/erss/textes/pagespersos/bourigault/TALN02-Bourigault.doc

CLAIR : Computational Linguistics And Information Retrieval group, University of Michigan (2003).

perun.si.umich.edu/clair/

Debili, F. (1982), *Analyse syntaxico-sémantique fondée sur une acquisition automatique de relations lexicales-sémantiques*, Thèse de doctorat d'état, Université de Paris XI, Centre d'Orsay.

Déjean H. (1998), *Concepts et algorithmes pour la découverte des structures formelles des langues*, thèse de l'Université de Caen.

Frath P., Oueslati R., Rousselot F. (2000), Identification de relations sémantiques par repérage et analyse de cooccurrences de signes linguistiques, In J.Charlet, M.Zacklad, G.Kassel & D.Bourigault, eds, *Ingénierie des Connaissances, évolutions récentes et nouveaux défis*, Paris, Eyrolles.

www-ensais.u-strasbg.fr/liia/publications/for00.ps

Lucas N. (2001), Étude et modélisation de l'explication dans les textes, Actes du Colloque "L'explication: enjeux cognitifs et communicationnels", Paris.

Rousselot F. (2002), LIKES (LIinguistic and Knowlege Engineering Station) : outil de traitement de corpus et d'aide à la construction d'ontologies.

www-ensais.u-strasbg.fr/liia/likes/likes.htm

Saussure F. de (éd. 1974), *Cours de Linguistique Générale*, Paris, Payot.

Vergne J. (2000), *Trends in Robust Parsing*, tutoriel du CoLing 2000, Nancy, Sarrebrück.

www.info.unicaen.fr/~jvergne/tutorialColing2000.html

Vergne J. (2001), Analyse syntaxique automatique de langues : du combinatoire au calculatoire (conférence invitée), Actes de TALN 2001, 15-29.

www.info.unicaen.fr/~jvergne/TALN2001_JV.ppt.zip

Vergne J. (2002), Une méthode pour l'analyse descendante et calculatoire de corpus multilingues : application au calcul des relations sujet-verbe, Actes de TALN 2002, 63-74.

www.info.unicaen.fr/~jvergne/TALN_2002/TALN2002_JVergne.doc.pdf

Zipf G. K. (1949), *Human Behavior and the Principle of Least Effort*, New York, Harper, réédition 1966.

Mise en place d'un Système de Recherche d'Informations en vietnamien

Bao-Quoc HO, Jean-Pierre CHEVALLET, Marie-France BRUANDET

Laboratoire CLIPS – IMAG
BP 53 – 38041 Grenoble Cedex 9
{Ho-Bao.Quoc;Jean-Pierre.Chevallet;Marie-France.Bruandet}@imag.fr

Résumé – Abstract

L'utilisation du traitement automatique de la langue naturelle (TALN) sur la recherche d'information (RI) donne des résultats positifs pour les langues européennes. En ce qui concerne les langues minoritaires comme le vietnamien, aucune étude n'a encore été réalisée dans le domaine de la Recherche d'Information. Dans ce travail, nous proposons une méthode d'indexation pour le Vietnamien en tenant compte des spécificités de cette langue. Dans cet article, nous présentons les spécificités du vietnamien d'un point de vue de la RI, puis nous présentons un outil d'analyse du vietnamien que nous avons mis au point pour la tâche de RI. Il est basé sur des règles de transformation. Nous présentons finalement les expérimentations que nous avons réalisées pour tester le premier système de recherche d'information dédié au vietnamien.

Application of Natural language processing (NLP) on information retrieval (IR) is a special issue of IR domain. There are positive results for European languages but for small languages like Vietnamese, researches are at their first stages. In this work, we propose an effective indexing expression for Vietnamese. We have studies specificities of Vietnamese language from the IR point of view. We have test the use of single word, compound word, noun phrase as indexing terms. We also present a Vietnamese analyser tool that we built, based on transformation rules. We finally present experiments that we carried out for this first Vietnamese Information Retrieval System.

Mots Clés – Keywords

Recherche d'information, Traitement Automatique de la Langue Naturelle en vietnamien
Information Retrieval, Vietnamese Natural Language Processing

1 Introduction

Un Système de Recherche d'Information (SRI) est un système d'assistance automatique à une recherche thématique d'informations contenues dans des documents. Il comporte deux phases principales : l'indexation et l'interrogation. Les SRI textuels doivent traiter les documents dans la phase d'indexation pour en extraire un index, capable de faciliter l'appariement entre la requête, support du besoin d'information de l'utilisateur, et le document qui contient l'information recherchée. Le texte peut être vu comme un signal brut et être traiter en tant que tel, c'est à dire de manière purement statistique. Cependant, les techniques de base du traitement de la langue peuvent être mises à profit pour améliorer la précision du SRI. Dans la phase d'indexation, les outils linguistiques comme des analyseurs de partie du discours, peuvent être utilisés pour extraire des termes d'indexation qui présentent au mieux le contenu des documents. Dans la phase d'interrogation, il peu être intéressant de construire des thésaurus qui permettent l'expansion des requêtes pour en préciser le contenu et rendre ainsi l'appariement requête-document plus fiable. De manière générale, l'efficacité de l'impact des traitements linguistiques dans un SRI, par opposition à des traitements purement basés sur des statistiques, n'a pas encore été clairement établie. Les résultats actuels sont néanmoins encourageants (W.A Woods et al.,)

De manière concrète, un SRI est confronté aux problèmes de variation linguistique comme la variation morphologique, la variation lexicale, la variation syntaxique et la variation sémantique. Le but des traitements linguistiques est alors de normaliser ces variations. L'application du TALN à la RI a été étudiée principalement sur les langues européennes. Pour les langues minoritaires comme le vietnamien, les études sont inexistantes et le domaine de la RI dans ces langues en est à ses premières étapes. Dans cet article, nous présentons les spécificités du vietnamien sous le point de vue de la RI. Nous présentons également les expérimentations que nous avons réalisées. Pour ces expérimentations nous avons du tout construire à partir d'aucun existant hormis le système RI lui même qui a pu être récupéré de l'anglais. En particulier, nous avons du construire un analyseur vietnamien de partie du discours, un extracteur de syntagmes, et une collection de test constituée de requêtes associées à leur résolution.

Le choix d'un terme d'indexation basé sur le mot isolé par des espaces, est adapté aux langues européennes, mais ce choix est complètement à revoir pour les langues asiatiques, en particulier pour le vietnamien, car les mots sont composés d'unités plus petites, séparés par des espaces, qui sont plus de l'ordre d'une syllabe. La reconnaissance même de ces mots est donc une difficulté propre à ces langues. Nous proposons alors dans ce travail, de faire précéder la phase d'indexation par une phase de reconnaissance des mots et de leur catégorie grammaticale (partie du discours), suivit d'une extraction de syntagmes.

Cet article est organisé en trois parties, nous présentons d'abord les spécificités du vietnamien puis un outil d'analyseur vietnamien basé sur la méthode des règles de transformation de Brill que nous avons réalisé et enfin, les premières expérimentation d'un SRI vietnamien.

2 Spécificités du vietnamien

2.1 Les spécificités du mot

Le vietnamien est une langue monosyllabique qui utilise un alphabet latin avec des accents sur les voyelles pour créer de nouvelles tonalités comme Ă, Â, Ê, Ô, O, U. En effet, le vietnamien possède six tons différents qui modifient le sens des mots, par exemple : ma (fantôme), má (joue), mà (dans le mot « mà còn » : encore), mǎ (tombe), mā (apparence), mă (semis).

Une phrase en vietnamien se compose de mots eux-mêmes composés d'unités linguistiques séparées par un espace. Chaque unité est une chaîne de caractères qui peut ne pas avoir de signification propre. Pour faciliter la présentation, nous utilisons les caractères "[]" pour désigner les mots composés d'au moins deux unités linguistiques. Par exemple, dans le mot [khủng hoảng] (la crise), la première unité n'a pas de sens quand elle est utilisée seule, et la deuxième signifie « affolé ». Cette construction des mots est particulière au vietnamien et pose un problème pour la segmentation d'un texte en mots. C'est une difficulté pour le traitement automatique du vietnamien en général, et pour la recherche d'information (RI) en particulier.

Les mots vietnamiens sont morphologiquement invariables, il n'y a donc pas besoin de phase de lemmatisation dans l'indexation. Il existe pourtant des suffixes et préfixes en vietnamiens, mais ils ne sont pas couramment utilisés. Par exemple, le préfixe « sự » transforme un verbe en substantif : [lựa chọn] (choisir) et [sự [lựa chọn]] (choix), tandis que le suffixe « hóa » réalise l'opération inverse : [tin học] (informatique) et [[tin học] hóa] (informatiser).

La catégorie d'un mot ne peut pas être reconnue grâce à des marqueurs morphologiques comme dans les langues ayant la variation de morphologie. Nous devons déterminer la catégorie d'un mot uniquement par le contexte où ce mot apparaît. Par exemple :

thành công (nom : réussite) của dự án đã tạo tiếng vang lớn

La réussite du projet crée un grand écho.

Bạn đã **thành công** (verbe : réussir) trong nghiên cứu khoa học

Vous avez réussi dans la recherche scientifique.

Buổi diễn rất **thành công** (adjectif : réussi).

Le concert est réussi.

Le mot « thành công » dans la première phrase est un substantif, dans la deuxième c'est un verbe et il est un adjectif dans la troisième. En recherche d'information, si l'on veut indexer les documents vietnamiens par un « sac de mots », on doit faire face au problème de la segmentation du texte en mots significatifs. Les tâches comme la lemmatisation, l'extraction des racines (racinisation) deviennent inutiles. Nous posons alors comme hypothèse que l'utilisation des mots composés et des syntagmes nominaux est plus efficace pour l'indexation de documents vietnamiens.

Environ 80 % des mots vietnamiens sont des mots comportant deux unités linguistiques. Partant de ce constat, nous avons réalisé une première expérimentation qui permet de mesurer l'importance du choix du terme d'indexation. En effet, le choix classique pour les langues européennes consiste à prendre pour terme d'indexation, les unités linguistiques séparées par des espaces. Ce choix est correct pour ces langues ou l'unité linguistique correspond la plupart du temps à un mot. Par exemple en français, les mots composés non liés par un tiret (comme "pomme de terre"), sont assez peu courants, et le fait de ne pas les reconnaître à l'indexation influe peu sur la qualité du SRI. Par contre, en vietnamien, sachant que dans leur majorité, les mots sont composés d'au moins deux unités, nous pouvons nous attendre à une influence sur la qualité des réponses du SRI.

2.2 Les spécificités du syntagme nominal

La structure d'un syntagme nominal vietnamien est un problème encore discuté par les linguistes de cette langue. Nous avons donc adopté un point de vue raisonnable qui est le suivant : un syntagme nominal vietnamien possède trois parties; une partie principale considérée comme la tête du syntagme, une partie qui précède la tête et une partie qui suit la tête. La partie "tête du syntagme" est un substantif simple ou un substantif composé. La partie précédant la tête contient des déterminants et la partie suivant la tête contient des modificateurs. Par exemple :

Partie précédente	La tête	Partie suivante	
tất cả các cuôn	sách	Tin học	Tous les livres d'informatique
loại	Khoai tây	vừa mua	La pomme de terre achetée
	Tin học	Công nghiệp	Informatique industrielle

Table 2. Les syntagmes nominaux vietnamiens

La partie qui suit la tête est une partie complexe. Elle peut contenir des mots de catégories différentes comme substantif, adjetif, verbe, pronom, numéro et même un syntagme complet. Par exemple :

Sách thi u nhi (SUBC SUBC) = livre d'enfant

Sách c  (SUBC ADJ) = ancien livre

Ph ng đ c (SUBC VERB) = salle de lecture

Xe m ri l m (SUBC NUM) = v hicule num ro quinze

Nh  ch ng t i (SUBC PROM) = notre maison

Sách b o thư vi n đ t mua (SUBC [SUBC VERB]) = les livres et journaux r serv s par la biblioth que

Il n'existe pas actuellement d'étude complète du syntagme nominal en vietnamien. Il n'existe donc pas de structure prédéfinie d'un syntagme nominal basée sur les catégories grammaticales comme on peut en trouver dans les langues européennes. La méthode classique par automates d'états finis utilisée dans ces langues n'est donc pas applicable ici. D'autre part, nous n'avons pas de corpus d'apprentissage suffisamment vaste pour mettre en oeuvre la

méthode dite « memory-based ». C'est pour ces raisons, que nous avons choisi la méthode de règles de transformation présentée dans cet article également pour l'extraction des syntagmes.

3 Analyse automatique du vietnamien

Nous avons construit un analyseur vietnamien qui permet l'étiquetage des mots selon leur catégorie grammaticale (parties du discours) et ensuite permet l'extraction des syntagmes nominaux. Nous avons utilisé la méthode d'apprentissage des règles de transformation d'Eric BRILL (Brill, 1995) modifiée par Radu Florian et Grace Ngai pour construire un analyseur syntaxique du Vietnam. Nous présentons les idées principales de cette méthode.

3.1 La méthode d'apprentissage de règles de transformation

La méthode d'apprentissage des règles de transformation, proposée par Brill en 1995, est une des meilleures méthodes d'apprentissage automatique des règles syntaxiques (Florian et al, 2001). Elle est souple et puissante pour les tâches de traitement automatique de la langue comme l'étiquetage (part of speech) (E. Brill, 1995), parseur (E. Brill, 1996) et l'extraction de syntagme (L. Ramshaw and M. Marcus, 1994)

L'idée principale de cette méthode est de considérer que le traitement des tâches de segmentation, d'étiquetage et d'extraction de syntagme, sont des tâches de classification. Par exemple, la segmentation des mots peut être vue comme une tâche de classification des unités qui les composent selon trois classes : D le début d'un mot, M le milieu d'un mot et F la fin d'un mot. L'étiquetage est une classification des mots par un ensemble de catégories et l'extraction de syntagmes est considérée comme une segmentation en mots de taille plus grande.

Cette méthode comporte deux phases : la phase d'apprentissage et la phase d'application. Dans la phase d'apprentissage, un corpus d'apprentissage doit être construit manuellement pour permettre la construction automatiquement des règles de transformation. Ces règles extraites sont ensuite utilisées pour traiter un nouveau corpus.

La production des règles se fait par rapport à un ensemble de modèles. Chaque modèle fixe le contenu d'une règle à l'aide de métasymboles indiquant que l'on se réfère au mot (word), à sa classe (chunk) ou bien à sa catégorie grammaticale (pos pour part of speech). Ces métasymboles sont associés à une position relative à la position courante d'application de la règle. Une règle, lorsqu'elle est appliquée, change la classe du mot courant de son application. Par exemple la règle :

```
word_-1="la" word_0="voiture" pos_-1=ART pos_0=VERB => pos_0=SUB
```

exprime que si on trouve le mot *voiture* catégorisé comme un verbe, précédé par le mot *la* catégorisé comme un article, alors on peut changer la catégorie du mot courant *voiture* en substantif. La classification concerne ici l'attribution d'une catégorie grammaticale. Cette règle peut correspondre au modèle suivant :

```
word_-1 word_0 pos_-1 pos_0 => pos
```

Ces modèles s'organisent selon quatre types : les modèles concernant le mot, ceux concernant le contexte du mot, ceux concernant la catégorie du mot, et ceux concernant le contexte de la catégorie d'un mot. Ces types de modèles vont être présentés plus en détails dans les parties suivantes. On utilise une fonction d'évaluation pour valider l'intérêt d'une règle.

De manière générale, un corpus d'apprentissage est un ensemble de mots associé à une classe qui dépend du type d'apprentissage en cours. Nous appelons *classe correcte* la classe qui a été affectée manuellement à chaque mot du corpus. Nous appelons *classe inférée* la classe qui est produite par l'application des règles. Dans notre cas, la classe peut être une catégorie grammaticale (partie du discours), ou une classe de construction des mots (début, milieu, fin de mot). Les règles sont apprises sur les exemples puis appliquées à l'ensemble du corpus d'apprentissage de manière à minimiser les erreurs. Le processus s'arrête lorsqu'aucune amélioration ne peut plus être apportée à la production des classes.

Au cours de l'apprentissage, la classe inférée est comparée à la classe correcte du corpus pour évaluer l'intérêt d'une règle. Pour cela, une fonction *Score* évalue l'impact d'une règle sur le corpus d'apprentissage. Ce score est calculé sur la différence après application de cette règle, entre le nombre de bonnes corrections de classes et le nombre de mauvaises modifications de classes par rapport à la classe correcte du corpus d'apprentissage :

Etant donnée une règle r , un mot m , $x(m)$ est la classe courante associée à m , $x(r, m)$ est la classe associée au mot m après avoir appliqué la règle r .

$$\text{Score}(r) = \sum \text{good}(r) - \sum \text{bad}(r)$$

Où $\text{good}(r) = \begin{cases} 1 & \text{si } (x(m) \text{ est incorrect}) \wedge (x(r, m) \text{ est correct}) \\ 0 & \text{sinon} \end{cases}$

$$\text{bad}(r) = \begin{cases} 1 & \text{si } (x(m) \text{ est correct}) \wedge (x(r, m) \text{ est incorrect}) \\ 0 & \text{sinon} \end{cases}$$

L'algorithme de la phase d'apprentissage est le suivant :

1. Création d'un état initial des classes du corpus d'apprentissage
2. Parcours de l'ensemble des modèles
3. Pour un modèle, parcours du corpus unité par unité
4. Production d'une nouvelle règle
5. Appliquer cette règle à la copie courante du corpus
6. Calculer le score de cette règle basée sur la fonction d'évaluation
7. On ne garde cette règle que si son score est positif
8. Pour un modèle, choisir la règle ayant le score plus haut, ajouter à l'ensemble des règles apprises
9. Appliquer la règle choisie à la copie courante du corpus
10. Si les modèles ne produisent plus de nouvelles règles, arrêter la phase d'apprentissage, sinon répéter tout le processus depuis l'étape 2

L'étape initiale 1 consiste à affecter à chaque mot, la classe statistiquement la plus probable. Cela signifie que pour chaque mot du corpus, on comptabilise les classes possibles à partir des classes correctes, puis on initialise la classe inférée à la classe la plus fréquente. A l'étape 4, la production d'une règle consiste à instancier le modèle pour extraire du corpus à la position donnée, les éléments (mots, classe ou pos) en partie gauche de la règle. Dans la partie

gauche, la classe correspond à la classe inférée dans l'état actuel du corpus. La partie droite correspond à la classe correcte. Pour un modèle et une position dans le corpus, il n'y a donc qu'une seule règle produite. On peut noter que cet algorithme est très coûteux car il va parcourir tout le corpus pour produire une seule règle. Il y a donc autant de parcours de corpus que de règles apprises. Dans la suite, nous présentons plus en détail des modèles et des règles utilisées dans cette méthode.

Un modèle a la forme suivante :

$$\langle \text{trigger}_1 \rangle \dots \langle \text{trigger}_n \rangle \langle \text{classe-courante}_1 \rangle \dots \langle \text{classe-courante}_n \rangle \\ \Rightarrow \langle \text{classe-correcte}_1 \rangle \dots \langle \text{classe-correcte}_n \rangle$$

où

$\langle \text{trigger}_i \rangle$ est une unité linguistique ou une catégorie grammaticale
 $\langle \text{classe-courante}_i \rangle$ est la classe associée au trigger i à un instant donné
 $\langle \text{classe-correcte}_i \rangle$ est la classe correcte du trigger i .

Par exemple, la tâche d'extraction des syntagmes nominaux consiste en la classification d'un mot appartenant à une des trois classes suivantes : B (begin, début d'un syntagme), I (in, milieu d'un syntagme) et O (out, dehors d'un syntagme). Les modèles dans ce cas ont la forme suivante :

$$\text{word_0 chunk_0} \Rightarrow \text{chunk} \quad (1)$$

$$\text{word_}-1 \text{ word_0 chunk_}-1 \text{ chunk_0} \Rightarrow \text{chunk} \quad (2)$$

$$\text{pos_0 chunk_0} \Rightarrow \text{chunk} \quad (3)$$

$$\text{pos_}-1 \text{ pos_0 pos_1 chunk_0} \Rightarrow \text{chunk} \quad (4)$$

Dans un modèle de règle, *word* représente une unité linguistique, *chunk* est la classe, et *pos* est la catégorie grammaticale (partie du discours, part of speech). Le chiffre qui suit représente la position de l'élément dans le corpus par rapport à la position courante de construction d'une nouvelle règle. Dans le premier exemple, 'word_0' est le mot examiné, 'chunk_0' désigne la classe courante du mot examiné et 'chunk' est la classe correcte pour le mot examiné.

Dans le deuxième exemple, 'word_-1' est le mot précédent du mot examiné, 'word_0' est le mot examiné, 'chunk_-1' est classe courante du mot précédent, 'chunk_0' est classe courante du mot examiné et 'chunk' est la classe correcte pour le mot examiné. En d'autres termes, ce modèle concerne des règles de contexte d'un mot. Le troisième exemple est un modèle de règles concernant la catégorie grammaticale du mot examiné. Et la dernière concerne les règles de contexte pour les catégories des mots

Si l'on utilise les modèles de l'exemple ci-dessus, on peut trouver des règles suivantes :

$$\text{word_0} = \text{« le »} \text{ chunk_0} = \text{‘O’} \Rightarrow \text{chunk} = \text{‘I’} \quad (1')$$

$$\text{pos_}-1 = \text{“PRO”} \text{ pos_0} = \text{“SUBC”} \text{ pos_1} = \text{“ADJ”} \text{ chunk_0} = \text{‘O’} \Rightarrow \text{chunk} = \text{‘I’} \quad (4')$$

La règle (1') issue du modèle (1) : signifie que si on trouve le mot « le » classifié 'O' (en dehors d'un syntagme) alors, il faut changer sa classe à 'I' (dans le contexte d'un syntagme).

Le règle (4') issue du modèle (4) : signifie que si la catégorie du mot précédent est un pronom, la catégorie du mot examiné est un substantif et la catégorie du mot suivant est un adjectif et le mot examiné est classifié 'O' alors, il faut changer sa classe en 'I'.

Dans la phase d'application, le corpus à traiter est parcouru une seule fois pour appliquer les règles de la phase d'apprentissage dans l'ordre du score. Dans des parties suivantes, nous présentons les expérimentations que nous avons réalisées pour le vietnamien.

3.2 Corpus d'apprentissage

Cette partie décrit la construction d'un corpus d'apprentissage pour la construction de règles de catégorisation grammaticales et d'extraction de syntagmes nominaux. Nous avons catégorisé un corpus de 37 000 mots et extrait les syntagmes nominaux manuellement avec l'aide de linguistes. Nous avons ensuite décomposé ce corpus en deux parties : la première partie contient 90% du corpus et est considérée comme le corpus d'apprentissage; la deuxième partie (10% du corpus) est le corpus de test.

3.3 Outil d'extraction des syntagmes nominaux

L'état initial des classes inférées du corpus d'apprentissage est calculé par la statistique de répartition des classes, comme décrit précédemment. Pour diminuer le temps d'apprentissage, nous avons initialisé l'ensemble normalement vide des règles apprises par un ensemble de règles construites manuellement. Puis ces règles ont été appliquées une première fois sur le corpus. Dans cette phase, si nous utilisons seulement l'information statistique, les erreurs représentent 8,7% de la taille du corpus. Lorsque nous combinons la statistique avec les règles prédefinies, les erreurs tombent à 6,8% de la taille du corpus. Le format du corpus d'apprentissage est le suivant : chaque ligne est composé d'un mot, de la catégorie de ce mot, et de sa classe (B,I,O).

Par exemple :

Nguyễn nhân (la cause)	SUBC I
là (est)	VERB O

Nous avons adapté au vietnamien l'ensemble des modèles proposés par Brill. En particulier, nous avons supprimé les modèles qui concernent un contexte trop large (≥ 3 mots précédents ou ≥ 3 mots suivants) pour diminuer le temps de la phase d'apprentissage. Cet outil est programmé en Delphi sur Windows. Le test donne une précision de 70% et avec un temps d'apprentissage de 10 heures 35 minutes. Nous pensons que la précision peut être améliorée en augmentant la variété des catégories grammaticales et la taille du corpus d'apprentissage. Grâce à cet outil, nous pouvons expérimenter l'impact de l'indexation des documents vietnamien par les syntagmes nominaux. L'utilisation de syntagmes nominaux pour la recherche d'information nécessite un changement de modèle d'indexation : le modèle vectoriel n'est plus applicable car la correspondance entre syntagmes doit s'apparenter à de la similitude d'arbre et non plus à des distances entre des vecteurs de termes. Pour réaliser ces

expérimentations, il faut donc mettre au point un nouveau modèle et construire un nouveau système de correspondance. Nous présentons donc dans la suite les résultats pour la recherche d'information utilisant les termes composés mais avant la phase de construction de ces syntagmes, donc toujours basé sur le modèle vectoriel de RI.

4 Expérimentations de la RI vietnamienne

4.1.1 *La collection de test*

Nous avons construit une collection de test pour pouvoir réaliser des mesures de qualité d'indexation. Une collection de test en RI est constituée d'un ensemble de documents, d'un ensemble de requêtes, et les solutions à ces requêtes, c'est à dire l'ensemble des documents jugés manuellement pertinents pour les requêtes. La collection est constituée d'articles en vietnamien sélectionnés à partir de journaux (*La jeunesse*) de l'an 2000 qui proviennent de l'Institut Linguistique Vietnamien de Hanoi. Cette collection contient 10 750 articles et a une taille brut de 23 Mo ; elle utilise le code des caractères TCVN3 utilisé dans la plupart des documents électroniques vietnamiens. A partir de ce corpus de documents, nous avons proposé 14 requêtes et nous avons évalué manuellement la pertinence avec les documents du corpus. Comme il est difficile de parcourir manuellement toute la collection, nous avons réalisé une première indexation des unités linguistiques avec SMART avec la pondération ltc. Pour toutes les requêtes, la liste des 20 meilleurs documents a été examinée manuellement pour construire la liste des réponses pertinentes.

Cette collection de test est à la fois un outil de mesure précieux et unique pour la langue vietnamienne. Cette collection nous a donc permis de mettre en place les expérimentations que nous décrivons ci-après.

4.1.2 *Les méthodes*

Nous avons réalisé cette expérimentation selon trois méthodes d'indexation : unigramme, bigramme et bigramme combiné avec un lexique. La notion d'unigramme correspond aux unités linguistiques. Comme nous l'avons décrit précédemment, ces unités ne se placent pas au niveau du mot de la langue. C'est pour cette raison que nous avons testé une méthode de regroupement des unités en couples. Nous avons finalement utilisé une ressource linguistique (un lexique) d'environ 30 000 items pour réaliser l'indexation. C'est le système SMART qui a été utilisé pour les tests avec une mesure de pondération des termes par la méthode ltc.

4.1.2.1 *Unigramme*

Dans cette première expérimentation, nous avons indexé la collection en utilisant une unité linguistique, l'unigramme, comme terme d'indexation. Le résultat est faible (précision moyenne de **0.3636**) mais il nous sert de base pour mesurer l'amélioration des résultats obtenus par les autres méthodes.

4.1.2.2 *B-gramme*

L'utilisation des bigrammes comme termes d'indexation, sous la forme de couples d'unités linguistiques, permet de retrouver l'équivalent du mot dans les langues européennes. Pour

cela, nous avons parcouru le texte de gauche à droite en extrayant tous les bigrammes possibles. Par exemple, étant donnée une phrase ABCDE, les bigrammes extraits sont AB, BC, CD, DE. Cette méthode, bien que plus adaptée à la langue vietnamienne, produit beaucoup de bruit dans l'ensemble des termes d'indexation. Par exemple : dans le cas du mot composé « công nghệ thông tin » (technologie d'information), ce dernier est découpé en les bigrammes suivants : « công nghệ » (technologie), « nghệ thông » (sans sens), et « thông tin » (information). La précision moyenne est néanmoins légèrement augmentée et vaut **0.3778**, ce qui prouve à notre avis, l'intérêt d'un découpage en mots plus correct.

4.1.2.3 Bigramme combiné un lexique

Pour enlever les termes d'indexation non significatifs produits par la méthode des bigrammes, nous avons utilisé un lexique constitué d'environ 30 000 items pour ne garder que les bigrammes existant dans le lexique. Cette méthode a donné une très nette amélioration, puisque la précision moyenne est de **0.5625**.

	Uni-gram	Bi-gram	Bi-gram + lexique
Précision moyenne	0.3636	0.3778	0.5625
% amélioration		1%	20%

Table 1. Pourcentage d'amélioration

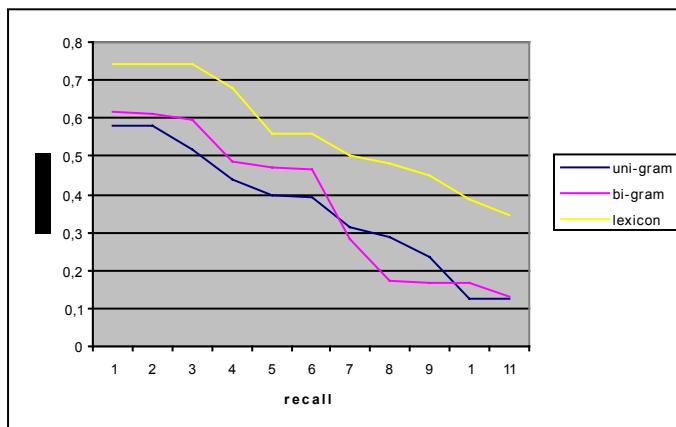


Figure 1. Courbe rappel - précision

Nous constatons que l'utilisation des bigrammes combinés avec un lexique est, pour l'instant, la meilleure stratégie d'indexation. Ce résultat semble donc indiquer que dans cette langue, le découpage en mots corrects est très important pour la tâche de recherche d'information. La technique du lexique a ses limites, car seuls les mots appartenant au lexique sont indexés. Or, dans la production de texte, il y a une part non négligeable de production de termes nouveaux. C'est même souvent sur ces mots nouveaux qu'une recherche d'information est pertinente pour l'utilisateur.

5 Conclusions

Nous avons étudié les spécificités du vietnamien du point de vue de la RI et réalisé les premières expérimentations d'un système de RI vietnamien. De plus, nous avons construit un

analyseur vietnamien pour catégoriser les mots et pour extraire des syntagmes nominaux qui seront le support des étapes suivantes pour l'indexation par les syntagmes nominaux.

La première étape du traitement de la langue vietnamienne est terminée et donne des résultats d'extraction satisfaisant. C'est la première fois qu'existent des outils automatiques de traitement de cette langue. Ils sont suffisamment généraux pour être utilisés dans un autre contexte. Nous avons spécialisé l'étape d'extraction des syntagmes pour la tâche de RI. Les expériences ont été menées dans le cadre du modèle vectoriel de RI. Il reste néanmoins un travail de modélisation pour construire un modèle de recherche d'information capable d'appréhender de manière correcte une indexation à base de syntagmes. Ce modèle doit ensuite être mis en œuvre de manière efficace pour que le temps de correspondance soit raisonnable pour le lancement de tests d'évaluation. Nos premières proposition se basent sur un modèle de dérivation de termes dans une logique terminologique.

Références

- A. Armpatzis, T. Tsoris, C. H. Koster, and T. P. Van Der Weide. (1998), "Phrase-based information retrieval. *Information Processing and Management*, 34(6) :693-707
- A. Arampatzis et al.,(2000), "Linguistically Motivated Information Retrieval. *Encylopedia of Library and Infoamtion Science*, Marcel Dekker, Inc., New York, Basel.
- E. Brill. (1996), "Recent Advances in Parsing Technology", chapter *Learning to Parse with Transformations*. Kluwer.
- E. Brill. (1995), "Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging". *Computational linguistique*, 21(4):543-565.
- G. Salton and M.J. McGill. (1983), "Introduction to Modern Information Retrieval". *McGraw-Hill, NewYork*, New York.
- Gómez et al. "Information Retrieval with Conceptual Graph Matching". In proceeding of the International DEXA Conference on Database and Expert Systems, 2000
- J.P. Chevallet and Hatem Haddad. (2001), « Proposition d'un modèle relationnel d'indexation syntagmatique : mise en oeuvre dans le système iota », in *INFORSID 2001, Genève-Martigny*, pp465, 483, 2001
- Lại Thị Hạnh. (2002), "Trích cụm danh từ tiếng Việt nhằm phục vụ cho các hệ thống tra cứu thông tin đa ngôn ngữ", *Memoire de Mastère du Université Nationale du Vietnam à HCM Ville*.
- P. Palmer. (1990), « Etude d'un analyseur de surface de la langue naturelle, application à l'indexation automatique des textes ». Thèse de doctorat, Université Joseph Fourier
- Nguyễn Hữu Quỳnh. (2001), "Ngữ Pháp Tiếng Việt", *Nhà xuất bản từ điển bách khoa*.
- Nguyễn Kim Thản. (1997), "Nghiên cứu ngữ pháp tiếng Việt". Nhà xuất bản khoa học xã hội.

Radu Florian, Grace Ngai. (2001), “Multidimension Transformation-Based Learning”. *Fifth Workshop on Computational Language Learning*.

L. Ramshaw, M. Marcus, (1999), *Natural Language Processing Using Very Large Corpora*, chapter Text Chunking Using Transformation-based learning. Kluwer

W.A. Woods et al., (2000), “Linguistique knowledge can improve information retrieval”. *In Sixth Annual Applied Natural Language Processing Conference, pages 262-267.*

Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens

Thi Minh Huyen Nguyen (1), Laurent Romary (1) et Xuan Luong Vu (2)

(1) LORIA
BP 239, 54506 Vandoeuvre lès Nancy
nguyen@loria.fr, romary@loria.fr
(2) Centre de Lexicographie du Vietnam
N° 67/4A, Ly Thuong Kiet Str., Hanoi, Vietnam
vuluong@vietlex.com

Résumé – Abstract

Dans cet article, nous discutons de la construction des jeux d'étiquettes pour l'analyse morpho-syntaxique du vietnamien, en prenant en compte les spécificités linguistiques de cette langue. Cette construction est inspirée du modèle MULTTEXT^(*) dans le but de s'orienter vers les applications multilingues ainsi que la réutilisabilité des jeux d'étiquettes. Nous allons finalement décrire une expérimentation sur l'étiquetage lexical des textes vietnamiens en utilisant QTAG (Mason et Tufis, 1998), un étiqueteur probabiliste indépendant des langues.

This paper discusses part of speech (POS) tagset construction for Vietnamese by considering linguistic specificities of this language. We take into account the schema as defined in the MULTTEXT^(*) model, so as to account for possible multilingual applications as well as the reusability of defined tagsets. Finally we describe our experiments on tagging Vietnamese texts using QTAG (Mason and Tufis, 1998), a language independent probabilistic tagger.

Mots Clés - Keywords

partie du discours, corpus de textes, étiquetage morpho-syntaxique, MULTTEXT, normalisation, QTAG

MULTTEXT, part-of-speech (POS), POS tagging, QTAG, standardization, text corpus

^(*) Multilingual Text Tools and Corpora <http://www.lpl.univ-aix.fr/projects/multext/>

1 Introduction

Chaque mot d'une langue appartient potentiellement à une ou plusieurs parties du discours selon son contexte d'utilisation. L'étiquetage lexical consiste à attribuer une étiquette morpho-syntaxique pour chaque mot dans un texte. Cette tâche est essentielle pour tout traitement ultérieur comme l'analyse syntaxique, sémantique ou même pragmatique d'une langue.

La notion de mot dans une tâche d'étiquetage lexical ne correspond pas nécessairement à un mot traditionnel en raison de la segmentation aveugle des textes sans information syntaxique ou sémantique. Un mot traditionnel peut être divisé en plusieurs unités ou morphèmes (dans le cas d'amalgames ou de mots composés, par exemple). Au contraire, plusieurs mots en séquence peuvent être groupés en une seule unité : des locutions, des noms propres composés, des numéros composés, des mots composés, etc. En fonction de la définition des unités lexicales et/ou de l'application, les descriptions des classes et des étiquettes morpho-syntaxiques peuvent inclure un ou plusieurs traits comme la catégorie syntaxique, le lemme, le genre, le nombre, etc. Dans (Przepiórkowski et Woliński, 2003), les auteurs proposent une nouvelle classification purement morpho-syntactique. Ils défendent l'idée que plusieurs jeux d'étiquettes polonais existants sont naïfs linguistiquement du fait de l'adoption directe, sans analyse critique préalable, des classes traditionnelles de parties du discours, ce qui cause un manque de réutilisabilité. (Tufis, 1998) a proposé un jeu d'étiquettes à deux couches dans le but de réduire les coûts de temps et de mémoire dans le processus d'étiquetage exploitant un jeu de plus de 700 étiquettes.

Il existe aujourd'hui différents outils pour l'étiquetage morpho-syntaxique, ainsi que d'immenses ressources de corpus annotés destinées à des traitements variés dans nombreuses langues. Les projets Treebank (<http://www.cis.upenn.edu/~treebank/home.html>) sont des exemples de création de larges corpus annotés. Cela suppose également l'existence de définitions variées d'unités lexicales et de jeu d'étiquettes selon l'objectif visé. Dans le cadre de Multext (Ide et Véronis, 1994) et de Multext-Est (Erjavec et al., 1996), des jeux d'étiquettes ont été définis pour une dizaine de langues avec un haut niveau de consensus au sujet de la structure de description.

Aussi se posent les questions cruciales du caractère réutilisable de ces ressources linguistiques pour un nombre croissant d'applications, leur réutilisation combinée dans un contexte multilingue, et l'adaptation d'un outil à d'autres langues. De multiples projets ont vu le jour dans cette perspective : l'évaluation des outils, la normalisation et la représentation des structures de description morpho-syntaxique (Ide et Romary, 2001).

Dans le cas des textes vietnamiens, le travail d'étiquetage est une tâche nouvelle et difficile pour les informaticiens, essentiellement du fait du désaccord sur la classification linguistique traditionnelle des mots au sein de la communauté linguistique. A ce jour il n'existe aucun standard reconnu pour les catégories des mots en vietnamien. Notre recherche vise deux objectifs principaux : en premier lieu créer des outils et des ressources linguistiques pour les applications de traitement automatique des textes vietnamiens, mais aussi assurer la disponibilité de ces outils pour les linguistes travaillant sur le vietnamien.

Après un bref état de l'art dans le domaine de l'étiquetage, nous présentons les spécificités linguistiques importantes du vietnamien dans le but de définir un jeu d'étiquettes. Pour la construction de jeu d'étiquettes, nous nous sommes volontairement basés sur le modèle

Multext intrinsèquement dédié aux applications multilingues. Ce jeu d'étiquettes sera évalué grâce à l'étiqueteur QTAG (Mason et Tufis, 1998).

2 Travaux antérieurs d'étiquetage lexical

2.1 Méthodologie et évaluation

Un état de l'art complet de l'étiquetage morpho-syntaxique est présenté dans (Paroubek et Rajman, 2000). L'étiquetage lexical s'effectue usuellement en trois étapes : segmentation du texte en unités lexicales, accès à un lexique pour récupérer toutes les étiquettes possibles pour chaque mot, et désambiguïsation pour attribuer l'étiquette correcte à chaque mot. Il existe deux approches principales pour la tâche de désambiguïsation : les méthodes à base de règles et les méthodes probabilistes.

Les méthodes à base de règles exploitent un ensemble de règles grammaticales pour résoudre le problème de l'étiquetage. Les méthodes non supervisées utilisent des contraintes produites par les linguistes et un lexique contenant pour chaque mot ses étiquettes possibles. Un tel étiqueteur s'apparente à un parseur (par ex. les systèmes GREYC et SYLEX présentés dans l'évaluation GRACE - Grammaires et Ressources pour les Analyseurs de Corpus et leur Evaluation). Les méthodes supervisées construisent les étiquettes et les règles de transformation à partir de corpus étiquetés manuellement. L'étiqueteur de Brill est l'exemple le plus connu de telles méthodes. A chaque mot est ensuite attribuée l'étiquette la plus fréquente selon le lexique. Enfin, les règles de transformation servent à la correction itérative de l'étiquetage préalable.

Les méthodes probabilistes (dont les systèmes utilisant le modèle de Markov caché) utilisent la distribution de probabilité sur l'espace des associations possibles entre les séquences de mots et les séquences d'étiquettes. Cette distribution est produite à partir du corpus d'apprentissage étiqueté ou non. La désambiguïsation entre les étiquettes d'un mot s'opère par le choix de la séquence d'étiquettes qui maximise la probabilité conditionnelle de l'association avec la séquence de mots courante. Ces méthodes reposent sur deux hypothèses. La probabilité d'association entre un mot et une étiquette est entièrement conditionnée par la connaissance de l'étiquette. Ensuite la probabilité d'occurrence d'une étiquette est conditionnée par la connaissance d'un nombre fixe d'étiquettes voisines.

La performance des systèmes d'étiquetage se mesure le plus souvent par le taux de précision (au niveau des mots) qui dépend fortement de la nature et de la taille du jeu d'étiquettes. La plupart de ces systèmes ont une performance supérieure à 90%. Le meilleur résultat obtenu dans l'évaluation du projet GRACE était de 97.8%.

2.2 Aspect de normalisation dans le domaine de l'étiquetage lexical

De gros efforts ont porté sur la normalisation des données, des outils et des ressources linguistiques pour favoriser leur réutilisabilité pour les recherches et les applications en traitement des langues à base de corpus. Multext en est un exemple significatif. Dans le cadre de ce projet, un modèle morpho-syntaxique a été développé en vue de l'harmonisation de

l'étiquetage de corpus multilingue ainsi que de la comparabilité des corpus étiquetés. Multext défend l'idée que dans un contexte multilingue, des phénomènes identiques devraient être encodés de manière similaire dans chaque langue pour faciliter les traitements dans des applications diverses (alignement automatique, extraction de terminologie multilingue, etc.).

Un principe du modèle est de séparer les descriptions lexicales qui sont en général stables, et les étiquettes du corpus. En ce qui concerne les descriptions lexicales, le modèle utilise deux couches : un noyau commun pour des catégories communes et une couche privée contenant des informations additionnelles qui sont propres à une langue ou aux applications particulières. Une solution de compromis pour les jeux d'étiquettes morpho-syntaxiques dans le noyau commun est un jeu de 11 étiquettes : Nom (N), Verbe (V), Adjectif (A), Pronom (P), Déterminant (D), Adverbe (R), Adposition (S), Conjonction (C), Numéral (M), Interjection (I), Résidu (X). L'information optionnelle de la deuxième couche est présentée par les paires attribut-valeur (structure de traits). Par exemple, un nom commun singulier est présenté par N[type = common gender = masculine number=singular case=n/a] (forme contracté Ncms-).

Or, il est évident que pour couvrir une plus grande variété de langues, il est nécessaire de présenter plus de flexibilité dans ce cadre fondamental. L'étude que nous présentons sur le jeu d'étiquettes vietnamien prouve qu'en effet quelques catégories peuvent ne pas convenir aux objets linguistiques réels de cette langue. Du point de vue de la normalisation, ceci signifie qu'une étape ultérieure serait soit de décrire une ontologie entière des catégories (comme suggéré par Farrar et al., 2002), soit d'enregistrer la variété de descripteurs possibles à travers des langues en construisant un enregistrement de méta-données (cf. Ide et Romary, 2001). Ces deux options ne sont pas nécessairement contradictoires puisque les catégories de données élémentaires peuvent se diriger aux noeuds dans l'ontologie, laissant comparer des jeux d'étiquettes à travers des langues ainsi que dans une langue donnée. De ce fait, il est important de considérer que pour une langue comme le vietnamien, un schéma d'annotation peut se fonder sur plusieurs couches de granularité d'étiquettes, et ceci devrait être pris en compte. La section suivante présente une telle stratégie, qui pourrait mener en particulier à une proposition d'un ensemble de références de descripteurs pour le vietnamien, dans le contexte du comité d'ISO TC37/SC4 (<http://www.tc37sc4.org>).

3 Définition d'un jeu d'étiquettes pour le vietnamien

Le vietnamien est une langue isolante, dans laquelle chaque mot a une forme unique et ne peut pas être modifié par dérivation ou flexion. Les relations grammaticales ne se manifestent pas par la flexion mais par l'ordre des mots. La classification de parties du discours n'est pas donc morphologiquement évidente.

3.1 Lexique

La langue vietnamienne a une unité spéciale appelée "tiếng" qui correspond en même temps à une syllabe du point de vue phonologique, à un morphème du point de vue syntaxique, à un sémantème du point de vue de la structure du mot, et à un mot du point de vue des constituants de la phrase. Il y a trois types de "tiếng" :

1. "tiéng" ayant un sens réel comme *sông* (rivière), *núi* (montagne), *đi* (aller), *đứng* (tenir debout), *nhớ* (se souvenir), *thương* (aimer/avoir pitié), ..., peut constituer à lui seul un constituant de phrase complet du point de vue syntaxique et sémantique. Ce type de mot est appelé **mot lexical**.
2. "tiéng" comme *nhưng* (mais), *mà* (que), *tuy* (bien que), *nên* (alors) ..., qui ne peut pas être un constituant de phrase à lui seul, mais qui est utilisé pour composer un constituant de phrase lexical, est appelé **mot outil**.
3. "tiéng" qui vient du chinois comme *son* (montagne), *thuỷ* (eau), *gia* (maison), *bất* (négation) ... ou qui a un sens flou et qui est en général combiné avec une autre syllabe comme *cô* (*xe cô* - véhicule), *đẽ* (*đẹp đẽ* - beau), *vé* (*vui vẻ* - joyeux)... permet de créer des mots et peut parfois être utilisé comme un mot.

Parmi les diverses définitions du mot en vietnamien, les linguistes sont parvenu à un accord et considèrent comme un mot la plus petite unité ayant un sens spécifié et une structure stable, et utilisée pour composer des constituants de phrase. Le lexique vietnamien contient : (i) Des **mot simples** ou de mots monosyllabiques correspondant aux catégories 1 et 2 de "tiéng". (ii) Des **mot complexes** qui ont plus d'une syllabe. Il existe principalement trois types de combinaison des syllabes : redoublement phonétique (par ex. *trắng*/blanc - *trắng* *trắng*/blanchâtre), coordination sémantique (par ex. *quần*/pantalon, *áo*/chemise - *quần* *áo*/vêtement) et composition sémantique (par ex. *xe*/véhicule, *đạp*/pédaler - *xe* *đạp*/bicyclette). On note aussi l'existence de mots composés dont les syllabes ne sont plus reconnaissables (*bò nông*/pélican). (iii) Enfin, **des expressions figées et des locutions**, qui sont généralement considérées comme des unités lexicales.

A cause de la grande fréquence des mots composés, la segmentation des textes en vietnamien est assez compliquée.

3.2 Jeu d'étiquettes

Le problème de classification des catégories grammaticales en vietnamien est toujours en débat au sein de la communauté linguistique (Hữu Đạt et al., 1998 ; Diệp et Hoàng, 1999 ; Cao, 2000). La difficulté vient de l'ambiguïté des rôles grammaticaux de nombreux mots. La mutation catégorielle verbe-nom est bien fréquente (sans aucune variation morphologique). Généralement les déterminants peuvent être utilisés comme des noms. Même les prépositions (par ex. *trên*/sur, *trong*/dans) jouent parfois le rôle de noms (*trên*/le supérieur, *trong*/l'intérieur), etc. Dans cette section nous présentons notre approche pour définir un jeu d'étiquettes conforme aux applications de Traitement Automatique des Langues (TAL).

Les catégories grammaticales reflètent des oppositions diverses dans le système syntaxique. Le principal critère pour la définition de notre jeu d'étiquettes est donc la distribution syntaxique. Nous devrions avoir un important jeu d'étiquettes pour refléter exactement toutes les relations syntaxiques. Cependant, plus le jeu d'étiquettes est important, plus la tâche d'annotation est difficile. Aussi, nous avons besoin d'un compromis pour parvenir à un jeu d'étiquettes assez précis et de taille acceptable. Nous commençons avec un petit jeu d'étiquettes généralement admis dans la littérature (cf. Uỷ ban KHXH, 1983), et figurant dans différents dictionnaires vietnamiens. Ce jeu comporte neuf catégories: Nom (N), Verbe (V),

Adjectif (A), Pronom (P), Adjonction (J), Conjonction (C), Interjection (I), Mot de Modalité (E) et Résidu (X). Notre tâche est alors de définir un nouveau jeu d'étiquettes en subdivisant chacune de ces catégories à l'aide d'étiquettes plus spécifiques.

En nous inspirant des principes de construction du modèle Multext, nous avons élaboré des spécifications lexicales pour le vietnamien dans un schéma comparable à ce modèle. Les différences du jeu d'étiquettes ci-dessus avec celui de Multext sont les suivantes : les numéraux et les déterminants de Multext se retrouvent dans la catégorie des noms du vietnamien, les adpositions de Multext se retrouvent dans la catégorie des conjonctions du vietnamien ; la catégorie des adjonctions du vietnamien contient des adverbes et des adjonctions de noms (noms pluralisant) de Multext ; la classe des modaux est propre au vietnamien. Pour rester conforme à Multext, nous avons ajouté la catégorie des numéraux à ce jeu d'étiquettes. Cette classe comprend les cardinaux et les ordinaux de la classe des noms et les adjonctions de noms de la classe des adjonctions. Par conséquent, seuls les adverbes restent dans la classe des adjonctions. Nous n'avons pas récupéré les classes Déterminant et Adposition du modèle Multext à cause de particularité de la grammaire vietnamienne. En définitive, nous obtenons le jeu d'étiquette de premier niveau suivant : Nom (N), Verbe (V), Adjectif (A), Pronom (P), Adverbe (J), Conjonction (C), Numéral (S), Interjection (I), Particule Modale (M), Résidu (X). Ci-dessous, nous présentons des spécifications lexicales de base pour chaque catégorie, fondées sur les combinaisons lexicales possibles.

- **Nom** : Seul l'attribut **type** (commun ou propre) dans Multext est approprié pour le vietnamien. Par contre, nous définissons de nouveaux attributs dont les valeurs sont entre crochets : **collective** [yes (*cây cối*/végétation), no (*cây*/plante)], **sense** [*object* (*nhà*/maison), *plant* (*lúa*/riz), *animal* (*mèo*/chat), *human* (*học sinh*/élève), *material* (*sát*/fer), *abstract* (*tình cảm*/sentiment), *fact* (*sự*/fait), *space* (*trong*/intérieur), *time* (*ngày*/jour), *senses* (*màu*/couleur), *style* (*giáo sư*/professeur)], **countable** [*absolute* (*cái*/chose¹), *partial* (*bàn*/table), no (*nhân dân*/peuple)], et **unit** [*classifier* (*cái*/le-un), *collective* (*bộ*/ensemble), *exact measurement* (*lit*/litre), *rough measurement* (*năm*/poignée)].
- **Verbe** : Comme les verbes du vietnamien ne sont pas flexionnels, aucun attribut défini dans Multext n'est approprié ici. Nous avons donc créé de nouveaux attributs propres au vietnamien : **transitive** [yes (*viết*/écrire), no (*ngủ*/dormir)], **sense** [*psychology* (*tin*/croire), *discourse* (*nói*/dire), *direction* (*lên*/s'elever), *movement* (*chạy*/courir), *existence* (*mất*/perdre), *transformation* (*trở thành*/devenir), *volition* (*muốn*/vouloir), *acceptation* (*bi*/subir), *comparison* (*bằng*/égalier), *residual* (*viết*/écrire)]. De plus, il existe un verbe spécial "*là*/être", qui est son étiquette lui-même.
- **Adjectif** : Le seul attribut pour les adjectifs du vietnamien est **type** [*quality*, *quantity*] (par ex. *đẹp*/jolie, *cao*/haut).
- **Pronom** : L'attribut principal intéressant de cette catégorie est son **type**, car il n'y a pas de cas, de genre ou de nombre dans la grammaire vietnamienne. L'ensemble de

¹ "cái" est un nom classificateur. Par exemple : *cái/chose bàn/table* = la table, *một/un cái/chose bàn/table* = une table, *cái/chose này/cette* = cette chose

valeurs appropriées à cet attribut est : personal (par ex. *tôi/je, chi/vous-soeur*), temporal (par ex. *bây giờ/maintenant*), demonstrative (par ex. *đây/ici, này/ce*), quantitative (par ex. *tất cả/tout, bây nhiêu/autant*), predicative (par ex. *thế/cela*), et interrogative (par ex. *ai/qui, gì/quoi*).

- **Adverbe** : Les adverbes du vietnamien sont des mots outils très importants pour exprimer le temps, changer le degré du prédicat, etc. d'une phrase. Nous définissons un nouvel ensemble de valeur pour l'attribut **type** : time (par ex. *đã/temps passé, sẽ/temps futur*), degree (par ex. *rất/très, quá/trop*), continuation/similarity (par ex. *cũng/aussi, vẫn/toujours*), negation (par ex. *không/ne pas*) et imperative (par ex. *hãy/particule impératif, đừng/ne pas faire*). De plus, un autre attribut est ajouté : **position** [pre, post] (par ex. *đã/déjà, ròi/déjà*).
- **Conjonction** : Cette catégorie emploie l'attribut **type** avec deux valeurs : subordinating (par ex. *của/de, do/à cause de, để/pour*) et coordinating (par ex. *và/et, nhung/mais, néu ... thi/si ... alors*).
- **Numéral** : Les valeurs de l'attribut **type** de cette classe sont : cardinal (*một/un*), ordinal (*nhất/premier*), adjunct (*những/pluralisant*).
- **Interjection**: Aucun attribut n'est associé à cette classe.
- **Mot de modalité** : Nous distinguons deux types de mots dans cette catégorie : particle correspondant aux mots ajoutés à une phrase afin de changer son intensité, et copulative correspondant aux mots ajoutés au début ou à la fin d'une phrase afin d'exprimer le sentiment de l'orateur.
- **Résidu** : Ce sont les unités lexicales et les expressions qui n'ont pas de classification spécifique.

D'autres traits pourront être ajoutés pour des objectifs différents (information sur la forme composée ou sur la forme redoublée, etc.). Pour ces spécifications lexicales, nous assignons 48 étiquettes à un jeu de deuxième niveau. Dans la section suivante, nous présentons un étiqueteur stochastique de textes vietnamiens avec deux jeux d'étiquettes définis.

4 Processus de l'étiquetage

Aucune recherche au sujet de l'étiquetage de partie du discours vietnamien n'a été publiée à ce jour. Nous avons démarré le travail en construisant un lexique vietnamien, dans lequel on associe chaque mot à ses étiquettes possibles dans les jeux d'étiquettes mentionnés précédemment. Comme nous l'avons discuté (section 3.2), le jeu d'étiquettes du deuxième niveau que nous avons choisi est un compromis afin d'éviter un jeu d'étiquettes trop grand. Pour valider ce choix, nous appliquons ce jeu d'étiquettes sur des corpus dont nous vérifierons à terme la distribution syntaxique. Un outil pour étiqueter automatiquement un corpus avec un jeu d'étiquettes donné est indispensable. Nous nous servons de l'étiqueteur QTAG à cette fin.

QTAG est un étiqueteur stochastique indépendant des langues. Il crée le lexique, le jeu d'étiquettes, les probabilités lexicales et contextuelles à partir du corpus manuellement

étiqueté. Grâce à cette base d'apprentissage, l'étiqueteur peut trouver les étiquettes possibles avec leur fréquence pour les assigner à chaque unité lexicale dans un nouveau corpus déjà segmenté. Si la recherche d'une unité dans le lexique échoue, l'étiqueteur essaie de lui trouver les étiquettes possibles par sa forme morphologique. Au pire des cas, cette unité se voit attribuer toutes les étiquettes existantes. Enfin, l'étiqueteur effectue la tâche de désambiguïsation en utilisant les distributions probabilistes apprises à partir du corpus.

On supprime le prédicteur morphologique de QTAG, puisque le vietnamien est une langue sans variation morpho-syntaxique. Nous nous concentrons maintenant sur la construction du lexique et du corpus d'apprentissage et puis évaluons les résultats obtenus.

4.1 Ressources langagières et corpus d'entraînement

En nous appuyant sur le Dictionnaire Vietnamien (Hoang Phe, 2002), nous construisons un lexique de 37454 unités lexicales, dont chaque unité a ses propres étiquettes. Ce lexique inclut des termes usuels du lexique de la vie quotidienne et des journaux, des termes fréquents en littérature, des termes dialectaux fréquemment utilisés, des termes scientifiques ou techniques dans les documents scientifiques populaires, des expressions usuelles, des syllabes spéciales seulement utilisées pour la composition des mots, et des abréviations d'usage courant. Le lexique est graduellement enrichi avec de nouveaux mots apparus dans les corpus traités.

Avant l'étiquetage proprement dit (manuel ou automatique), le premier pas est la segmentation, i.e. l'identification des unités lexicales dont la définition est donnée dans la section 3.1. Le vietnamien est monosyllabique, mais les mots composés sont fréquents. Cela ne permet pas une simple segmentation par les espaces dans un texte. Pour résoudre ce problème, nous avons adopté les automates d'états finis pour identifier des segmentations possibles pour chaque phrase (délimitée par des ponctuations). En pratique, la segmentation correcte la plus probable est le chemin le plus court dans le graphe. Dans le cas ambigu (plusieurs chemins de la même longueur), une intervention humaine est nécessaire. Cette solution simple s'avère efficace dans la plupart des cas. Quelques améliorations de cette méthode pourraient être faites dans le futur proche (par ex. l'identification des formes redoublées, la désambiguïsation utilisant l'information de partie du discours, etc.). Une autre approche de segmentation est présentée dans (Dinh Dien et al., 2001).

Ensuite, le corpus segmenté destiné à l'apprentissage de l'étiqueteur est manuellement annoté après le passage d'un outil d'étiquetage préalable. En vue de l'expérimentation, nous avons annoté un corpus de 74753 unités dont 63733 unités lexicales (à peu près de 10000 unités lexicales différentes, sans compter des ponctuations). Un cinquième de ce corpus est composé de textes journaux, le reste, de textes littéraires.

4.2 Evaluation

Notre étiqueteur modifié prend en compte le lexique construit (section 4.1). Nous avons entrepris 6 essais sur deux jeux d'étiquettes définis avec une taille croissante du corpus d'entraînement. Le texte restant dans le corpus manuellement étiqueté est employé pour le but d'évaluation.

Voici un exemple du résultat de l'étiquetage pour la phrase "**hòi / lên / sáu / , / có / lân / tōi / dã / nhìn / tháy / môt / búc / tranh / tuyêt / đẹp**" qui est traduite mot à mot en "*quand / monter / six / , / avoir / fois / je / déjà / regarder / voir / un / [classificateur] / image / extrême / beau*" (Lorsque j'avais six ans, j'ai vu, une fois, une magnifique image) :

```
<w pos="Nt"> hòi</w> <w pos="Vto"> lên </w> < w pos="Sc"> sáu </w>
    <w pos=","> , </w> <w pos="Vte"> có </w> <w pos="Nt"> lân </w>
    <w pos="Pp"> tōi </w> <w pos="Jt"> dã </w> <w pos="Vtx"> nhìn </w>
    <w pos="Vtx"> tháy </w> <w pos="Sc"> môt </w>
    <w pos="Nc"> búc </w> <w pos="No"> tranh </w>
    <w pos="Jd"> tuyêt </w> <w pos="Aa"> đẹp </w>
```

dans lequel : Nt - nom temporaire, Vto - verbe transitif de direction, Sc - nombre cardinal, Pp - pronom personnel, Jt - adverbe temporaire, Vtx - verbe transitif (résidu), Nc - nom de classificateur, No - nom d'objet, Jd - adverbe de degré, Aa - adjectif de qualité.

L'expérimentation confirme que, plus le corpus d'entraînement est volumineux, plus le résultat est précis. Le meilleur taux de précision pour le jeu d'étiquettes du premier niveau est d'environ 94% (9 étiquettes lexicales et 10 ponctuations) avec un corpus d'entraînement d'environ 50000 unités lexicales (60000 unités au total). Pour celui du deuxième niveau, le meilleur taux de précision est d'environ 85% (48 étiquettes lexicales et 10 ponctuations) avec le même corpus d'entraînement. Sans utiliser le lexique construit ci-dessus, ces taux de précision sont à peu près de 80% et 60% respectivement. Une petite partie d'erreurs est due aux erreurs d'étiquetage dans les données d'entraînement. Bien que le résultat soit plutôt modeste en apparence, particulièrement au deuxième niveau, il n'est pas décourageant car la taille du corpus d'entraînement est encore très petite (50000 unités en comparaison aux centaines de milliers d'unités des corpus d'entraînement dans d'autres travaux sur l'étiquetage).

L'étiqueteur est disponible sur la page <http://www.loria.fr/equipes/led/outils.php> (vnQTAG) ainsi que les ressources linguistiques (lexique, corpus d'apprentissage).

5 Conclusions

Nous avons présenté notre travail sur l'étiquetage lexical du vietnamien. Puisque les chercheurs vietnamiens se sont très récemment impliqués dans le domaine de TAL, nous avons dû construire toutes les ressources linguistiques nécessaires et définir toutes les structures de données à partir de zéro. Néanmoins, nous tirons bénéfice de quelques avantages : beaucoup de méthodologies existantes pour l'annotation morpho-syntaxique et une forte conscience de la tendance de normalisation. Le jeu d'étiquettes défini pourrait être facilement ré-ajusté et étendu grâce à des descriptions lexicales. Ces descriptions sont en plus comparables à celles d'autres langues prises en compte dans le cadre du projet Multext. Avec l'aide de l'étiquetage automatique, nous pouvons facilement augmenter la taille du corpus annoté. Les résultats obtenus constituent une base pour d'autres recherches dans le domaine de TAL pour le vietnamien : analyse syntaxique, recherche d'information, alignement multilingue, traduction automatique, etc.

Références

- Uỷ ban Khoa học Xã hội Việt Nam (1983), *Ngữ pháp tiếng Việt (Vietnamese Grammar)*, Hanoi, NXB Khoa học Xã hội.
- Ide N., Véronis J. (1994). MULTTEXT: Multilingual Text Tools and Corpora. *Proceedings of the 15th International Conference on Computational Linguistics, COLING'94*, Kyoto, Japan, 588-92.
- Erjavec T., Ide N., Petkevic V., Véronis J. (1996) Multext-East: Multilingual Text, Tools and Corpora for Central and Eastern European Languages. *Proceedings of the First TELRI European Seminar*, 87-98
- Hữu Đạt, Trần Trí Dõi, Đào Thanh Lan (1998), *Cơ sở tiếng Việt (Basis of Vietnamese)*, Hanoi, NXB Giáo dục.
- Mason O., Tufis D. (1998), Tagging Romanian Texts: a Case Study for QTAG, a Language Independent Probabilistic Tagger, Proceddings of *First International Conference on Language Resources and Evaluation (LREC)*, Granada (Spain), 28-30 May 1998, p.589-596.
- Tufis D. (1998), Tiered Tagging, in *International Journal on Information Science and Technology*, vol. 1, no. 2, Editura Academiei, Bucharest, 1998.
- Diệp Quang Ban, Hoàng Văn Thung (1999), *Ngữ pháp tiếng Việt (Vietnamese Grammar, vol. 1-2)*, Hanoi, NXB Giáo dục.
- Cao Xuân Hạo (2000), *Tiếng Việt - mấy vấn đề ngữ âm, ngữ pháp, ngữ nghĩa (Vietnamese - Some Questions on Phonetics, Syntax and Semantics)*, Hanoi, NXB Giáo dục.
- Paroubek P., Rajman M. (2000), Etiquetage morpho-syntaxique, *Ingénierie des langues* (p. 131-150) Paris, HERMES Science Europe.
- Dinh Dien, Hoang Kiem, Nguyen Van Toan (2001), Vietnamese Word Segmentation, *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPERS2001)*, Tokyo (Japan), 27-30 November 2001, p. 749-756.
- Ide N., Romary L. (2001), Standards for Language Resources, *Proceedings of the IRCS Workshop on Linguistic Databases*, Philadelpia, 141-9.
- Farrar, S., W. D. Lewis, D. T. Langendoen (2002), An Ontology for Linguistic Annotation, *AAAI '02 Workshop: Semantic Web Meets Language Resources*.
- Hoàng Phê (2002), *Từ điển tiếng Việt (Vietnamese Dictionary)*, Vietnam Lexicography Centre, NXB Đà Nẵng.
- Przepiórkowski A., Woliński M. (2003, to appear), The Unbearable Lightness of Tagging* A Case Study in Morphosyntactic Tagging of Polish, *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, Budapest (Hungary), 13-14 April 2003.

Traitement automatique des langues minoritaires et des petites langues
organisée par Oliver Streiter (European Academy Language and law)

TALN 2003

Natural Language Processing for Amharic: Overview and Suggestions for a Way Forward

Atelach Alemu¹, Lars Asker² and Mesfin Getachew³

(1) Department of Information Science, Addis Ababa University, PO Box 1176,
Addis Ababa, Ethiopia
atelach@aaunet.aau.edu.et

(2) Department of Computer and Systems Sciences, Stockholm University and the
Royal Institute of Technology, Forum 100, SE-164 40 Kista, Sweden
asker@dsv.su.se

(3) Department of Information Science, Addis Ababa University, PO Box 1176,
Addis Ababa, Ethiopia
mesfin@sisa.aau.edu.et

Résumé

Nous donnons une vue d'ensemble des efforts qui ont été faits pour développer des systèmes de traitement de langage naturel pour l'Amharique, la langue officielle de l'Ethiopie. Nous discutons également brièvement un moyen possible d'établir les ressources de base (par exemple corps cru, corps morphologiquement, syntactiquement et sémantiquement annoté, base de données lexicologique de base, analyseur morphologique, tagger, et vérificateur d'orthographe) pour les applications utiles pour la technologie de langage. La quantité de travail exigée pour commencer à partir de zéro pour développer tous les aspects du traitement du langage naturel pour une nouvelle langue est énorme. Il y a en même temps un besoin pressant d'une variété d'applications, comprenant des correcteurs orthographiques de langues loceaux, des unités de traitement de texte, des systèmes de traduction automatique, des moteurs de recherche, etc... Pour que ces applications soient développées, l'existence des ressources automatisées de langue et un cadre bien développé pour la recherche dans ce secteur est exigée. "Tree-banks", "Part-of-speech taggers", les grammaires informatisées, et les lexiques, sont tous des parties nécessaires de cette structure. Il est essentiel que le travail commence avec le développement d'une fondation large, basée sur les corps, les lexiques et la morphologie. Le challenge est de développer ces ressources dans une façon qu'ils soient facilement étendu pour le couverage plein, et qu'ils soient réutilisables par d'autres outils ou applications. Il est mieux de mieux construire des ressources de fondation utiles à développer des outils généraux ou spécifiques, aussi que le logiciel d'application et des produits pour utilisateur final pour la langue en question. Ces fondations, si elles sont faites réalisable, deviendront la base pour les développements présents et futurs du travail relaté au traitement de langue naturel Amharique.

Abstract

We give an overview of efforts that have been made to develop natural language processing systems for Amharic, the official language of Ethiopia. We also briefly discuss a possible way to building basic resources (e.g. raw corpus, morphologically, syntactically and semantically annotated corpus, basic lexical database, morphological analyzer, tagger, and spelling checker/corrector) to useful language engineering applications of the language. The amount of work required to start from scratch in developing all aspects of natural language processing for a new language is huge. At the same time there is an urgent need for a variety of applications including local language spell-checkers, word processors, machine translation systems, search engines, etc. For these applications to be developed, the existence of computerized language resources and a well developed framework for research in this area is required. Tree-banks, Part-of-speech taggers, computerized grammars, and lexica, are all necessary parts of this framework. We see it as essential that this should begin with the development of a broad foundation based on corpora, lexicon and morphology. The challenge is to develop these resources in a way that they are easily extended for full coverage and make them reusable by any other tool or application as central component. It is better to build foundation resources useful to develop general and specific tools and also applications and end-user products in the language. These foundations, if they are made feasible, will become the base for present and future developments of work related to Amharic natural language processing.

Mots Clés

“Part-of-speech tagger”, Amharique, apprentissage automatique

Keywords

part-of-speech tagger, Amharic, machine learning

1 Background

Amharic is the official government language spoken in Ethiopia. It is a Semitic Language of the Afro-Asiatic Language Group that is related to Hebrew, Arabic, and Syrian. Amharic, the syllabic language, uses a script which originated from the Ge'ez alphabet (the liturgical language of the Ethiopian Orthodox Church). The language has 33 basic characters with each having 7 forms for each consonant-vowel combination. Unlike Arabic, Hebrew or Syrian, the language is written from left to right. Amharic alphabets, as Chinese alphabets differ from Arabic alphabets, are one of a kind and unique to Ethiopia (Child of the world, 1998).

According to the 1998 census in (Arthur Lynn's World Languages) Amharic is spoken widely through out different regions of Ethiopia: by over 17 million people as a first language and by over 5 million second language users. Some estimates indicate that Amharic is the mother-tongue of around 15 to 30 million Ethiopians.

Outside the country, this language is also spoken in Egypt, Israel and Sweden (AMHARIC: a language of Ethiopia, NY). Amharic, for instance, is spoken by over 40,000 Jews of Ethiopian origin in Israel (AMHARIC: a language of Ethiopia, NY) and the language has now become “One of Africa's latest-comers as a national language”.

Manuscripts in Amharic are known from the 14th century and the language has been used as a general medium for literature, journalism, education, national business and cross-communication. A wide variety of literature including religious writings, fiction, poetry, plays, and magazines are available in the language (Arthur Lynn's World Languages).

Processing Amharic language using computers has become common for more than a decade. A growing number of people these days use computer systems for processing the language. People, for instance, use computers for various purposes: doing document writing and correction, storage and retrieval of Amharic texts and databases. These have become feasible long ago and make the production of more and more documents (information) and databases to be made available in the language, in electronic form.

There are a number of Amharic word processing software packages available in the market. From our observation, the most widely used are the Power Geez, Visual Geez and Samawerfa. None of these, or any of the other Amharic software, supports language specific utilities like spell checking, grammar support, on-line thesaurus, etc. The absence of such word processing tools for Amharic language has thus made word processing activities using the language incomplete.

In addition to the above, there is a need to do remote information retrieval of documents written in the language, translate documents from other languages to Amharic and vice versa, consultation of Amharic electronic dictionaries and others.

The increasing use of the Internet intensifies the demand for some of the aforementioned needs. A prime example is the need to translate documents in other languages (e.g. English) to Amharic and vice versa. This need makes development of machine translation systems to be more useful. This is mainly attributed to the fact that many people (the majority) cannot make use of the huge information available on the Internet unless translated to local languages since they have no knowledge of foreign language.

Meeting such needs requires the presence of basic technology in the language (Amharic). Contrary to this, most basic working applications or language technologies are only available for few languages (e.g. English). It seems that no basic technologies are available for languages such as Amharic.

So, for the language to survive in the information age and help people in their development, it requires to develop basic technological tools for Amharic. The tools (which are required for the development of Amharic language processing software) include morphological analyzers, POS taggers, phrase recognizers, word sense disambiguation programs, parsers, translation aids and so on.

2 Current situation

Amharic is a language that, although it is the official language of Ethiopia and spoken by 15 -30 million people, suffers severely from lack of computational linguistic resources. Some attempts have been made to address these issues by developing limited prototypes that investigate the use of basic technologies for natural language processing of the language. Among them are design and development of Amharic word parser (Abiyot, 2000), automatic part of speech tagger for Amharic language (Mesfin, 2001), design and development of automatic morphological synthesizer for Amharic perfective verbs (Kibur, 2002), automatic morphological analyzer for Amharic by (Tesfaye, 2002), automatic sentence parsing for Amharic text (Atelach, 2002), Amharic speech recognition (Solomon, 2001), (Kinfe, 2002), Amharic OCR (Worku, 1997), (Ermias, 1998), (Dereje, 1999), (Million, 2002), (Nigussie, 2002), information retrieval (Zelalem, 2001), (Saba, 2001), (Bethlehem, 2002), stemming (Nega, 1999), machine translation (Sisay, Haller, 2003), and the development of an English-Amharic electronic dictionary (Sebsibe, 2001).

Although such work in developing basic technological tools for Amharic is encouraging, it is often very far from being complete and organized. For instance, almost all of the work described above was developed before linguistic foundations were created. This lack of basic linguistic resources brought a direct impact on the scope of research conducted by these and other researchers. The absence of no general sequence stated in developing Amharic language engineering applications was another obstacle in Amharic natural language processing. Still one weak point of such past work is that most of it was conducted for academic exercise. Once the work achieved their target, it seems that no follow-up is made to bring the research work to an operational level.

3 Description of some of the early work

3.1 Development of a word parser for Amharic verbs

Abiyot (2000) has tried to design and implement a word parser for Amharic verbs and their derivation. He designed a knowledge-based system that parses verbs, and nouns derived from verbs. He used root pattern and affixes to determine the lexical and inflectional category of the words. The study did not include the property of words at syntactic level. He experimented on a limited number of words (200 verbs and 200 nouns). The result showed that 86% of the verbs and 84% of the nouns were recognized correctly but the results are too specific to draw any general conclusions from due to the limited number of words used in the study.

3.2 Development of a part of speech tagger for Amharic

This work (Mesfin, 2000) was an attempt to develop a POS tagger for Amharic using a stochastic HMM, which model contextual dependencies. The tagger developed was a prototype and uses a page long text for both the training and test set. Mesfin's tagger extracts major word classes (Noun, Verb, Adjective, auxiliary, etc) and classes that are unique to the language. In total, a tag set containing 23 tags (derived form the page long text) was compiled. A lexicon of about 300 words was also developed manually. Mesfin's tagger does not support subcategory acquisition or constraints such as number, gender, polarity, tense case, and definiteness. It does not have a mechanism to estimate or guess POS tags for unknown words; instead it assigns "UNC" for such words. UNC here stands for unknown category. Since the lexicon built was so small, there was a high chance that the tagger would not find many of the words in the test set, and therefore assign them with this special tag. Apart from this, there were other limitations, including:

- Lack of better repository of real world knowledge (e.g. Treebank) useful for POS tagging
- Morphological complexity of the language and the lack of morphological analyzer
- Tag set was built from a page long text, thus the designed tag set should be refined and re-refined
- Tags were not designed to give morphological information
- Other constraints such as time and finance

3.3 Development of a Morphological analyzer for Amharic

Tesfaye Bayu (2002) developed and implemented a morphological analyzer of the language. In his study, he used two separate systems in order to develop the morphological system. The first system applies an unsupervised learning approach based on probabilistic models to extract morphemic components (prefix, stem and suffix) and construct a morphological dictionary. The second system, developed applying the principle of Auto segmental Phonology, was used to identify morphemic component of a stem such as consonantal root, vocalic melodies and CV-templates. And the test result showed that the first system was able to parse successfully 87% of words of the test data (433 of 500 words). This result corresponds to a precision of 95% and a

recall of 90%. Tested with 255 stems, the second system has also identified the morphemic compotes of 241 (or 94% of the) stems correctly.

Although morphological analysis is not sufficient to solve the problem of new words entirely (due to morphological ambiguity), combining statistical lexical co-occurrence techniques and morphological analysis could produce a better result.

3.4 Development of a sentence parser for Amharic texts

In this work (Atelach, 2002), an attempt was made to develop an automatic sentence parser for Amharic language using the probabilistic Inside Outside algorithm supplemented with chart parsing algorithm that implemented Probabilistic Context Free Grammars (Briscoe, Waegner, 1992). The prototype parser was developed for a constrained set of four word sentences in the language.

A sample corpus consisting of 100 sentences was used to generate phrase structure rules and to estimate probability values. The POS tagger developed by Mesfin (2001) was used to automatically tag the words in a sentence. The developed parser used the output of the tagger as an input.

The algorithms employed generate all possible parse structures for a given sentence, and calculates the probabilities of each possibility, to return the structure of the most probable parse. A database of PCFGs, which is used by the parser in determining the best parse was also designed and implemented.

Experiments were conducted in three phases, one on the training set, the second on the test set, and the third one on a set of four word sentences, different from the ones used as a training and testing set, obtained from different people. The first two sets were obtained from the sample selected for the purpose of the experiment.

Evaluation of the parser performance was made based on judgment of experts, and manual counts. In the study, only one parameter, the percentage of correct parse assignment as compared with the hand parses, was used to measure the performance of the parser.

The result achieved based on the first set of sample sentences was very high, 100% on the training set and approximately 96% on the test set. Before achieving such accuracy, the experiment was repeatedly done on both the training set and the test set with identifying errors and making corrections. This high accuracy was obtained partly due to the small number of words considered for the purpose of the experiment. Another reason was that all the sentences had identical constructions and the highest probability parses were almost always the correct ones.

Another set of sentences which were not included in the sample selected was parsed and the result obtained was 77%. All errors identified with the tagger were due to human made errors (therefore easily fixed) and untagged words. The untagged words problem was slightly handled by a module to guess the POSs of unknown words. The errors made by the parser were due to incomplete PGFGs (under generation) which were caused by the use of a very small corpus, and low probability assigned rules that could generate the correct parse for the misparsed sentences. In order to deal with the first problem, another set of 30 sentences (with a different construction

than that of the first set) was selected from the language and the PCFGs were re-estimated. This time the accuracy of the parser was found to be 81%. And the accuracy of the second test set remained to be 77%, this time the errors were due to low probability assignments.

Although the accuracy of the parser developed in this work was high, additional work is still needed to improve it to handle all kinds of Amharic sentences. In the tests conducted an average accuracy of 85% was obtained. This figure cannot be generalized to determine the performance of a PCFG parser for the language, due to the small sample size taken. Further work is still required to ensure that the PCFGs are representative enough and the probability values are accurate enough.

In this study, the PCFG extracted was limited to a small sample data for various reasons including: the lack of large corpora in the language annotated with POS tags and sentence parses, the expensive and time consuming process of generating the required amount of data, and the shortage of a hand parsed and tagged corpus.

The researcher had to manually tag and parse the sample selected. In the time available only a few sentences were selected and processed (i.e. tagged and parsed), grammar rules were generated and the rule probabilities were assigned. The rule probabilities were static, i.e. no dynamic probability re-estimation was used, also due to time constraints.

The sample taken for the study and the grammar rules generated cannot be taken to be a representative of the language, and future researches should be conducted using a larger set of corpus.

4 A Way Forward

Despite such limited attempts as those described above, there are still strong demands for the availability of basic technologies useful for natural language processing of Amharic (e.g. by researchers). Researchers' demand, for instance, seems escalating from time to time for necessary infrastructure suitable for automatic processing of the language, Amharic.

Thus, steps (or actions) need to be taken in building such basic infrastructure (e.g. raw corpus, morphologically, syntactically and semantically annotated corpus, basic lexical database, morphological analyzer, tagger, and spelling checker/corrector) for useful language engineering applications of the language (e.g. Amharic translation system). Building these fundamental technologies should not begin with advanced applications (e.g. machine translation). It rather should begin with the development of a broad foundation based on lexicon and morphology. It is better to build foundation resources useful to develop general and specific tools and also applications and end-user products in the language. These foundations, if they are made feasible, will become the base for present and future developments of works related to Amharic NLP works.

Among the basic technologies that should be made available as basic strategic priorities are the availability of a balanced corpus, a lexicon, a morphological analyzer and syntactic and semantic parsers. Another basic priority is the development of a part of speech tagger for Amharic. Thus, the challenge is to develop these resources in a way that they are easily extended for full

coverage and make them reusable by any other tool or application as central component (e.g. syntactic parsers, semantic parsers, and machine translation systems).

4.1 Creation of resources

One important resource is the existence of a balanced corpus. At the first level, a set of relevant linguistic data (useful for POS tagging) will be collected. Such data will be gathered mainly by hand, i.e. using expertise of linguists. The major aim in doing this is to develop an Amharic Treebank, a language resource that contains annotations of NL data at various levels, phrase, clause and sentence level (Megyesi, 2002).

Today ongoing projects on building Treebanks for several languages (e.g. Bulgarian, Chinese, French, Portuguese, Spanish, Turkish and so on) are common. Contrary to this, no Treebank exist or there are no on going projects on building Treebank for Amharic language. This study will address this issue as a side track.

Creation of such standard dataset, i.e. balanced text corpus or Treebank, for the application of automated methods of knowledge extraction will require a major effort (as such corpus has to be mainly built from scratch). We are currently in the process of compiling a balanced text corpus in collaboration with linguists.

Building a lexicon and a detailed tag set for the language are also among the major tasks. The presence of human experts is crucial in all works mentioned above.

In developing a POS-tagger, a machine learning approach will be employed in rather than the stochastic HMM approach. The reason for this is that statistical POS taggers have several drawbacks including the following

- They have difficulty in estimating small probabilities accurately from a limited amount of training data
- They are inflexible in the sense that they use the same basic strategy (e.g. bigram) for determining the POS of every word.
- The statistics, i.e. the lexical probabilities $p(\text{word} \setminus \text{tag})$, used in the tagging process are not based solely on sentences containing the word being tagged
- They use only a small amount of information in the training process(e.g. bigram)
- They require also large tables of statistics

To our knowledge, ML approaches to POS tagging of Amharic language has yet to be addressed. Thus, in our suggestion, ML approaches to tagging that avoid or address such problems will be considered. Constraint grammar using ILP and neural networks would be tried initially (take priority) and other techniques (such as Decision trees) would be explored in due course.

4.2 Steps ahead

We have identified the following steps to be taken:

- Collect, analyze, process large linguistic corpora of Amharic text with the aim to create a standard dataset, i.e. balanced text corpora or a Treebank, for the application of automated methods (e.g. POS tagging). In this, the study is aimed to build an initial Amharic Treebank. Effort will be made to make this corpus as useful as possible for different areas of linguistic research in the language. As mentioned above, this step has already been initiated.
- Manually tag words in the balanced corpus with appropriate lexical attributes (i.e. POS) with the help of experts. This allows making available a large tagged corpus (for training) in the language. Automatic or semi-automatic approaches will be tried for the processing and tagging of the balanced corpus.
- Create a lexical database by manually assigning tags to the words. This is a huge task. Thus automatic or semiautomatic approaches will be tried to create an appropriate lexicon from the tagged example corpus (training data)
- Determine a suitable or appropriate set of tags for the language. This requires considerable linguistic expertise.
- Determine probability distribution of features from the tagged corpus during the training phase, if found appropriate for the method to be followed for the study.
- Treat unknown words based on the technique to be selected to enable the tagger guess the appropriate POS of the words not known to the tagger. This might require developing and incorporating a stemmer or morphological analyzer to the tagger
- Create other knowledge required for the technique to be chosen (e.g. the constraint grammar rules in case of using ILP)
- Finally, build an efficient and accurate POS tagger (using ILP, Neural network or any other ML techniques found appropriate for the study)

5 Conclusions

Until a critical mass of resources such as corpus, lexicon, morphological analyzers and part-of-speech taggers have been made publicly available, it is not likely that natural language processing for Amharic can “take off” and reach a level higher than that of academic prototype systems. The fact that such resources would be made publicly available will further boost activities in the area by allowing researchers to benefit from the work of others and thereby to build more advanced tools without having to invent the wheel over and over again. We believe this to be the best chance for any language with limited computational linguistic resources to break the vicious circle of always having to start from scratch with research in the area of NLP.

References

Abiyot Bayou (2000), *Design and Development of Word Parser for Amharic Language*, Masters Thesis, Addis Ababa University.

Allen, James (1995), *Natural language Understanding*, Redwood City, Benjamin/ Cummings.

AMHARIC: a language of Ethiopia, http://www.ethnologue.com/show_language.asp?code=AMH

Arthur Lynn's World Language,
http://www.maps2anywhere.com/Languages/languages_-_amharic.htm

Briscoe, Ted and Waegner, Nick. (1992), Robust Stochastic Parsing Using the Inside Outside Algorithm. Proceedings of *AAAI: Workshop on Probabilistically Based Natural Language Processing Techniques*, San Jose.

Charniak, Eugene (1993), *Statistical Language Learning*, MIT Press, Cambridge, MA.

Charniak, Eugene (1997), *Statistical Techniques for Natural Language Parsing*, Brown university.

Child of the World : Amharic,
http://ourworld.compuserve.com/homepages/GenX_jt_mtjr/GenXAmharic.html

Chomsky, Noam (1957), *Syntactic Structures*, Netherlands, Mouton & Co.

Chomsky, Noam (1965), *Aspects of the Theory of Syntax*, MIT Press, Cambridge, Massachusetts.

Kay, M. (1979), Functional grammar, Proceedings of *the Fifth Annual Meeting of the Berkeley Linguistic Society*.

Mao, Yonghong (1997), *Natural Language Processing Module: Part of Speech tagging and Sentence Parsing, Laboratory Manual*, MIT.

Megyesi, B. (2002), About Treebanks, <http://www.speech.kth.se/%7Ebea/treebank.html>

Merlo, P. (1996), *Parsing with Principles and Classes of Information*, Dordrecht, Kluwer Academic publishers.

Mesfin Getachew (2001), *Automatic Part of Speech Tagging for Amharic: An Experiment Using Stochastic Hidden Markov (HMM) Approach*, Masters thesis, Addis Ababa University.

Sisay Fissaha and Haller, Johan (2003), Amharic verb lexicon in the context of Machine Translation, Proceedings of the *Workshop on Natural Language Processing of Minority Languages with few computational linguistic resources* held at TALN 2003.

Tesfaye Bayu (2002), *Automatic Morphological Analyser: An Experiment Using Unsupervised and Autosegmental Approach*, Masters thesis, Addis Ababa University.

TALN 2003

Amharic verb lexicon in the context of Machine Translation

Sisay Fissaha , and Johann Haller

Institute for Applied Information Sciences – University of Saarland

Martin-Luther-Str.14, D-66111, Saarbrücken, Germany

Tel +49-681-3895126, Fax +49-681-3895140

{sisay, hans}@iai.uni-sb.de

http://www.iai.uni-sb.de

Abstract

Cet article traite de trois problèmes concomitants reliés à la morphologie des verbes amhariques dans le contexte de la Traduction Automatique. L'amharique, tout comme d'autres langues sémitiques, présente un caractère morphologique complexe empêchant une description unique de ses caractéristiques majeures. Une brève analyse des différentes théories proposées pour la classification des verbes amhariques montre qu'un panachage de ces approches correspond au mieux aux besoins de l'application. De plus, l'analyse des différents phénomènes de dérivation suggère qu'un lexème constitué uniquement de consonnes est bien approprié aux spécifications du transfert lexical. Pour finir, bien que la plupart des difficultés morphologiques de l'amharique puissent être traitées par les mécanismes d'état fini et de modèle à deux niveaux, la dérivation réduplicative (qui implique un changement dans la courbe d'intonation des voyelles et dans le schéma de gémination) crée des difficultés contraignant la définition de modèles supplémentaires.

This paper discusses three related issues concerning the morphology of Amharic verbs in the context of Machine Translation. Amharic, like other Semitic languages, exhibits a complex morphological phenomenon defying a unified description of its important characteristics. A brief assessment of the different proposals made concerning the organization of Amharic verbs indicates that the amalgamation of the important characteristics of the different approaches better meet the requirements of the application at hand. Furthermore, analysis of the different derivational phenomena suggests that a lexeme consisting only of root consonants is well suited for specification of lexical transfer. Finally, although most of the complexities of Amharic morphology can be handled using the machinery of finite-state and two-level morphology, reduplicative derivation, which involves change in vowel melody and gemination pattern, poses difficulties forcing the stipulation of additional template forms.

Keywords

Amharic; lexicon; two-level morphology; machine translation; finite-state morphology; Semitic languages.

Amharique; lexique; morphologie à deux niveaux, traduction automatique; morphologie à état fini; langues sémitiques

1 Introduction

Amharic, which belongs to the Semitic family of languages, is one of the most widely spoken languages in Ethiopia. It has a complex morphology which makes complete listing of all surface word forms in the lexicon impossible. Thus the need for analysing Amharic words has long been recognized, which is reflected in some of recent attempts, such as Amharic word-processing (Daniel and Yonas, 1994), stemming algorithm (Nega, 1999), word parser (Abiyot, 2000), and parts-of-speech tagger (Mesfin, 2001). All these works may generally be characterized as providing shallow analysis which does not satisfy the requirements posed by machine translation systems.

The current study is part of a wider project that attempts to integrate Amharic into the CAT2 machine translation system. The first section of this paper presents some of the observations made regarding the organization of Amharic verbs in the course of developing Amharic morphological analyzer. In machine translation systems, the lexicon plays a significant role in the analysis and generation of text by providing the different levels of processes with morphological, syntactic and semantic information. In section two, we focus our analysis on the particular aspect of lexical entry, that is, the specification of the base lexeme in a way which allows compositionality of translation. Finite-state techniques and two-level morphology has been the main computational model for Arabic and other Semitic languages (Kay, 1987; Beesely 1996). The last section reports on the implementation of Amharic morphology under these frameworks.

2 Amharic Verb Classification

Amharic language, like other Semitic languages such as Arabic, exhibits the root-pattern morphological phenomenon. This is especially true of Amharic verbs, which rely heavily on the arrangement of the consonants and vowels in order to code different morphosyntactic properties. Therefore, identification of the most frequently occurring consonant and vowel patterns has been a logical starting point in most attempts to organize Amharic verbs into classes. For example, Bender and Fulas (1978) identified 11 major classes, each consisting of different number of subclasses (a total of 42 subclasses), for simple Amharic verbs using the criteria: consonantal skeleton, gemination pattern, occurrence of vowel other than *ä*, occurrence of initial *a* or *t*, presence of *w*, *y*, *h* in the root consonants, and identical consonants in sequence. Leslau (1995), Cohen (1978) and Dawkin (1969) also provide another classification of Amharic verbs on the basis of some of these criteria. One common characteristic of these approaches is that they propose a relatively large set of classes. Baye (1999) questions the above ways of organizing Amharic verbs. Specifically, he challenges the widely accepted belief that the number of consonantal radicals ranges from one to six. He claims that the base form of Amharic verbs consists of three radicals. Any deviation from this is to be accounted for through the process of root reduction or extension.

Although previous studies have been instrumental in identifying the generalizations underlying Amharic morphology, very little work has so far been done on the implementation aspect. In the current work, we try to point out the implication of the different proposals in implementation which we present in the next few paragraphs.

Formulation of the criteria mentioned above relies to some extent on the observation made on the surface forms as shown in Table 1 (Baye 1999). A root form takes different patterns for different morphosyntactic categories of which the most common ones are perfect, imperfect, imperative, jussive, gerundive, and verbal. We also see that the words differ with respect to

their surface realizations. The verb *sbr* is a tri-radical verb. All the three consonants appear in almost all morphological derivations of the verb. The verb *sma* is also postulated as tri-radical verb consisting of three consonants of which only the first two consonants appear in the surface form of the verb. The last consonant is a place holder for a lost laryngeal consonant whose absence is indicated on the surface by the change of vocalic pattern in the second radical, i.e. *ä* ->*a*, and also by the introduction of new radical *t* in the verbal form which does not appear in the base form. The third verb *AyY* lost the first and third radical hence exhibits an idiosyncratic surface pattern far different from the first and second verbs. The verb *mnzr* is a quadrilateral verb consisting of four consonants which appear in almost all surface forms of the verb resulting in different surface patterns. Although the last one *trg^wm* is also a quadrilateral verb consisting of four radicals exhibiting similar morphological processes as *mnzr*, it has been put in a different subclass because of the existence of a labiovelar consonant *g^w* which effects some idiosyncratic vocalic changes in some derivation of the surface forms. On the bases of their idiosyncratic properties all these verbs are grouped into different classes.

Radicals ¹	Stems	Perfect	Imperfect	Jussive	Gerund	Verbal
		CVCXVC	CVCC	CCVC	CVCC	CCVC
<i>sbr</i> – ‘break’	Underly.	<i>säbXär-</i>	- <i>säbr-</i>	- <i>sbär</i>	<i>säbr-</i>	- <i>sbär</i>
	Surface	<i>säbbär-</i>	- <i>säbr-</i>	- <i>sbär</i>	<i>säbr-</i>	- <i>sbär</i>
<i>sma</i> – ‘hear’	Underly.	<i>sämXää-</i>	- <i>sämA-</i>	- <i>smäA</i>	<i>sämA-</i>	- <i>smäA</i>
	Surface	<i>sämma-</i>	- <i>säma</i>	- <i>sma</i>	<i>Sämt-</i>	- <i>smat</i>
<i>qrY</i> – ‘remain’	Underly.	<i>qärXääY-</i>	- <i>qärY-</i>	- <i>qräY</i>	<i>qärY-</i>	- <i>qräY</i>
	Surface	<i>qärr-</i>	- <i>qär-</i>	- <i>qr</i>	<i>qärt-</i>	- <i>qrät</i>
<i>AyY</i> – ‘see’	Underly.	<i>AäyXääY-</i>	- <i>AäyY-</i>	- <i>AyäY</i>	<i>AäyY-</i>	- <i>AyäY</i>
	Surface	<i>Ayyä-</i>	- <i>ay-</i>	- <i>y</i>	<i>Ayt-</i>	- <i>ayät</i>
<i>mnzr</i> - ‘change’ money	Underly.	<i>mänäżXär-</i>	- <i>mänäżXär</i>	- <i>mänzär</i>	<i>Mänzär-</i>	- <i>mänzär</i>
	Surface	<i>mänäzzär-</i>	- <i>mänäzzär</i>	- <i>mänzär</i>	<i>Mänzär-</i>	- <i>mänzär</i>
<i>trgwm</i> -‘translate’	Underly.	<i>täräg^wXäm-</i>	- <i>täräg^wXäm</i>	- <i>tärg^wäm</i>	<i>tärg^wäm</i>	- <i>tärg^wäm</i>
		<i>täräggom-</i>	- <i>täräggum-</i>	- <i>tärgum</i>	<i>tärgum-</i>	- <i>tärgom</i>

Table 1: Bender and Fulas (1978): Simple verb forms

However, looking at the different underlying forms and the derivation process, one can see some regularities and relationship between the different verbs. The underlying stem form, e.g *säbXär-*, is obtained by intercalating the root consonants *sbr* and vowel patterns *ä* in the respective slots of the template. Note that the vowel *ä* does not constitute the vowel pattern rather it is inserted using the general epenthesis rules (using Xerox rule format) given in (1). It encodes the fact that Amharic does not allow consonant clusters at the beginning of a word.

- (1) *Cons* = Consonant set
 $[..] \rightarrow \wedge || \# . \text{Cons} _ \text{Cons};$

The underlying stem forms still contain abstract morphophonological elements (e.g. *X* for gemination) that need to be realized through the application of alternation rules. Gemination, for example, is handled using (2) which simply spreads the geminated consonant to the neighbouring slot *X* resulting in a sequence to two identical consonants.

1 The symbols A, and Y function as a placeholder for lost radicals. And the symbols C, V and X in the template forms indicate consonant, vowel, and gemination of the previous consonant respectively.

(2) $Cons = Consonant\ set$

$$X:C \Leftrightarrow C _ : ; \\ \text{where } C \text{ in } Cons ;$$

The verb *sämma* (*sämXäA-*), which lost the final radical *A*, introduces the consonant *t* in its verbal and gerund derivations and changes the quality of the vowel of the penultimate radical. This is taken care of using (3),

(3) $A \rightarrow t \parallel _ + Gerund$
 $\ddot{a} \rightarrow a \parallel _ A + Verbal$
 $A \rightarrow t \parallel _ + Verbal$
 $\ddot{a} \rightarrow a \parallel _ A + Perf$

The final radicals of *AyY* and *qrY* are mapped into the corresponding surface forms using (4).

(4) $Y \rightarrow t \parallel _ + Gerund$
 $Y \rightarrow t \parallel _ + Verbal$

(3) & (4) do not apply to the verb *sbr* as it does not have any lost radicals. However, because of its underlying pattern it is related with verbs like *AyY* and *qrY*.

The above sample alternation rules in turn show that these verbs have more common features. The apparent differences that one may observe on the surface form can be eliminated by using a few sets of rules and by postulating the right root forms.

The word forms like *mnzr*² do not have a related tri-radical form which has a semantically sound derivation. Therefore, it has been found necessary to postulate a quadrilateral template forms. The idiosyncratic vocalic pattern appearing in the surface forms of *trg^wm* (due to the labioveral *g^w*) can be generated using (5). *LabioCons* variable represents all the labiovelar consonants whereas *SimpCons* represents corresponding simple form.

(5) $LabioCons = \{l^w, m^w, r^w \dots\} \quad SimpCons = \{l, m, r, \dots\}$
 $LabioCons \rightarrow SimpCons o \parallel _ \ddot{a} ;$
 $LabioCons \rightarrow SimpCons u \parallel _ Cons ;$

We have also postulated template forms containing five radicals. There are very few verbs having five distinct radicals. Most verbs of this class contain duplicate radicals. As a result, one is tempted to derive these verbs from the corresponding tri-or quadri-radical forms. *bläCäläC* ('glitter') is an example verb of this class. Its perfect form is given by *-bläCälläC-*. Assuming a stem template form of *-bläC-*, the perfect form may be thought of as being derivative of *-bläC-* by copying the second and third radicals. Here we face some problems. On the one hand, *-bläC-* does not have any meaning as this form does not exist in the language. On the other hand, this derivational process seems to lack sound semantic basis which one usually finds in other similar derivational phenomena like reduplicative derivations, e.g. *säbbärä* ('he broke') and *säbabärä* ('he broke into pieces'). Due to the rarity of the verbs, and some practical reasons to be discussed in the next sections, we define five radical template forms.

² Baye (1999) claims that the second radicals of most quadriradical verbs have the feature [+continuant] and , hence, are predictable. However, this rule has some exceptions, e.g. *bätärräq*

In general, all the above suggest that some of the works (Bender et. al 1978; Leslau 1995) in Amharic morphology tend to provide a fine-grained classification whereas others (Baye 1999) postulate a highly abstract generalization about Amharic verb base forms. The position taken in this paper is an intermediate one. Some level of abstraction has been introduced in order to capture the generalization relating to commonly occurring lexical items at the same time postulating higher order template forms to avoid specification of highly abstract form.

3 Lexical entry specification for lexical transfer

CAT2 is a transfer-based machine translation system where the lexical transfer component constitutes the core of the transfer module. Lexical transfer is defined on a base lexeme defined in the lexicon. The fact that one can not list all word forms in the lexicon means one needs some recursive means of expressing meaning, i.e. compositional translation of the words along the line of compositional treatment of sentences. This raises an important question about the structure of the lexicon in general and the form of the lexical units which serve as bases for lexical transfer in particular.

A base form of an Amharic verb may be postulated at different levels of abstraction. However, the most plausible one, which is in line with the analysis provided in the previous section, is the one which considers root consonants as the base form of an Amharic verb. Hence assuming that the base form of Amharic verbs consists only of root consonants, an example lexical entry for the verb *sbr* would look like; {lex=*sbr*, ...}. Some of the simple and complex derived forms of this verb are given in Table 4.

Perfect	<i>säbbärä</i>	he broke
Imperfect	<i>y&säbr</i>	he was breaking or he will break
Causative	<i>assäbbärä</i>	he made someone break
Passive	<i>tesäbbärä</i>	it was broken
Reduplicative	<i>säbabbaärä</i>	he broke into pieces

Table 4: Derivational paradigm of *sbr*

The surface form *säbbärä* will consist of the template form (*CVCXVC*), root consonants (*sbr*), vowel pattern (*ä*), and the suffix morpheme (*ä*). While the root *sbr* determines the basic meaning for the word, the template form *CVCXVC* along with the vowel pattern provides additional morphosyntactic information such as aspect and tense. The suffix morpheme corresponds to the third-person-singular-masculine suffix pronouns. Compositional translation of the above verb would then mean translation of each of the above morpheme into the equivalent form of the target language. Using our definition of lexical entry, the base form would then be translated using a transfer rule of the form {lex=*sbr*} \Leftrightarrow {lex='break'}.

The lexeme *sbr* is stripped off all the grammatical opposition which makes it good candidate for specification of lexical transfer. It is abstract enough to include all concepts having to do with breaking. This analysis is in line with the idea introduced by Streiter(1996) that only those lexemes which correspond to the notional domain should appear in the lexical transfer. However, reducing all the surface forms to a single canonical form of *sbr* results in a significant loss of information. In order to avoid that, we should be able to encode their differences in some systematic way (e.g. use of semantic features). However, such process of abstraction should be done with care otherwise we may run into the problem of over-translation, in which compositional translation of the word results in some awkward translation which does not correspond to the meaning of the whole (Streiter 1996).

The suffix morpheme can be featurised and transferred into the target language whose synthesis component would then generate the required pronoun. Translation of the remaining morpheme (*CäCXäC*), however, is not straightforward. Direct transfer does not seem plausible as it does not have an equivalent form in the target language. Another option is to encode the syntactico-semantic information expressed by this morpheme.

The perfect form is used to render wide varieties of meaning. Typically it is used to express the meaning of past tense as in (6).

- (6) *lğ-u mästawät-u-n säbbär-ä-w*
 boy-DEF mirror-DEF-ACC breakPAST-SUBJ-OBJ
 ‘The boy broke the mirror’.

The tense may also vary depending on the context in which the perfect form is being used. In conjunctive construction, its tense depends usually on the tense of the main clause (7).

- (7) *ȝyebella anebbebe*
 ‘He read while he ate’.
ȝyebella yanäbbal
 ‘He reads while he eats’.

The imperfect form commonly expresses present and future tenses (8).

- (8) *lğ-u mästawät-u-n yʌ-säbr-al*
 boy-DEF mirror-DEF-ACC SUBJ-breakPRES-AUX
 ‘The boy will break the mirror’.

With the verb näbbärä, it can also be used to express habitual or durative action as in (9),

- (9) *lğ-u mästawät-u-n yʌ-säbr näbbär*
 boy-DEF mirror-DEF-ACC SUBJ-breakPRES AUX
 ‘The boy was breaking the mirror’.

In some cases, it may also deviate from the meaning licensed by the root form, e.g. *msl* ('resemble'), *yʌmäsl* ('as if')

- (10) *leba yʌ-mäsl*
 thief SUBJ-resembleIMPERF
 ‘as if he were a thief’

A similar conclusion may be made of jussive, gerund, imperative, and verbal forms. The template forms tend to behave as inflectional affixes having some basic meaning which occurs most frequently and with some exceptional deviations.

Complex derived forms include among others the passive, causative, reduplicative, and reciprocal. The derivational processes are either internal in which *CV* patterns are changed, or external where derivational affixes are attached to the simple derived forms discussed above. It may also involve a combination of internal and external changes. Some of the derivations in this class serve to express adverbial functions as the language has limited lexicalized adverbs. Others introduce wide varieties of modifications to the meaning expressed by the root. The passive, for example, is formed by prefixing the morpheme *tä* to the passive templates. Note

that passive template forms show some deviations from template forms we saw earlier. In addition to the usual passive meaning, the passive template forms are also used to turn transitive (11) into intransitive verbs (12), and express reflexive meaning (13).

- (11) *lğ-u mästawät-u-n mälläs-ä*
boy-DEF mirror-DEF-ACC returnPAST-SUB
'The boy returned the mirror.'
- (12) *lğ-u kä-tämari bät tä-mälläs-ä*
boy-DEF from-student house PASS-returnPAST-SUB
'The boy returned from school.'
- (13) *lğ-u tä-aṭṭäb-ä*
boy-DEF REF-washPAST-SUB
'The boy washed himself.'

There are a number of verbs which have *tä* as part of the basic stem (14).

- (14) *lğ-u abat-u-n täkättäl-ä*
boy-DEF his father followPAST-SUB
'The boy followed his father.'

There are two causative derivational morphemes *a* and *as*. Like the passive derivation, each of the causative derivational morphemes takes an overlapping set of template forms. The causative derivations affect the meaning of the basic verb by adding one or more element in the argument structure of the verb.

- (15) *lğ-u läbbäsä*
boy-DEF dressPAST-SUB
'The boy got dressed.'
lğ-u a-läbbäsä
boy-DEF caus-dressPAST-SUB
'The boy made (*someone*) dress.'

The reduplicative derivation mainly expresses such adverbial function as intensity, reduplication, repetition of action, etc. The reduplicative derivation involves mainly internal change. It introduces a new set of template forms in which the second radical will be duplicated in tri-radical verbs. Table 5 summarizes the different verb template forms for *sbr*.

Verb forms	Active	Passive	a-causative	as-causative	Reduplicative
Perfect	<i>CVCXVC</i>	<i>CVCXVC</i>	<i>CVCXVC</i>	<i>CVCXVC</i>	<i>CVCVXXVC</i>
Imperfect	<i>CVCC</i>	<i>CVCXVC</i>	<i>CVCC</i>	<i>CVCXC</i>	<i>CVCVXXC</i>
Gerund	<i>CVCC</i>	<i>CVCC</i>	<i>CCC</i>	<i>CVCXC</i>	<i>CVCVXC</i>
Imperative	<i>CCVC</i>	<i>CVCVC</i>	<i>CCC</i>	<i>CVCXC</i>	<i>CVCVXC</i>
Verbal	<i>CCVC</i>	<i>CVCVC</i>	<i>CCVC</i>	<i>CVCXVC</i>	<i>CVCVXVC</i>

Table 5. Template forms for derivational paradigm of sbr

These derivational forms are by no means exhaustive and may take different forms for other verb types. But they suffice to show several instances of syncretism that exist between the forms. The same template form (e.g. *CVCXVC*) serves different functions. In addition, the different derivational paradigms allow different degree of formalization. The simple derived forms may generally be associated with expression of aspect and tense which is relatively closed and amenable to formalization into semantic features. The meaning expressed by the

passive and causative forms show some degree of variation. However, due to the regular occurrence of the basic meaning, we can still benefit by positing some canonical form corresponding to the most typical meaning and handling exceptions with special rules. Reduplication, on the other hand, introduces quite a range of adverbial functions into the basic meaning of the root resisting any form of abstraction. In general, the possibility of decomposing verbs into constituting morphemes which obey the compositional treatment of the meaning of the word suggests that a lexeme consisting of root consonants is a good candidate for defining lexical entries on the basis of which lexical transfer operates.

4 Implementation

Two-level morphology has been the main computational framework in the field of computational morphology. Its dependence on concatenation operation has created, in its early stage, some difficulties in handling some non-concatenative phenomenon, like reduplication, and Semitic stem interdigitation. A number of attempts have been made in order to overcome these problems (Kay, 1987; Beesely 1996). Its Xerox finite state implementation carries a number of innovative ideas which circumvent these problems without deviating much from the basic underlying principles (Beesely 1996). The basic idea originates from Autosegmental approach to Semitic languages as it is formulated by McCarthy (1987). This is especially true of the treatment of consonantal roots, vowel melodies and the template forms as separate but interrelated entities. However, unlike the original work which represents each of these components in a different tier (multiple dimensions), it makes use of a linear representation.

As the name implies, two-level morphology involves specification of two levels (base form and surface form) that are related through rules. For Amharic, as in any Semitic language, the stipulation of base forms involves specification of the consonantal roots, template forms, and vocalic patterns and the associated morphosyntactic features. Figure 1 shows representation of the Amharic verb *säbbär-* using Xerox finite state regular language.

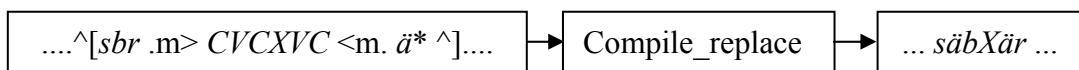


Figure 1 Regular expression for stem interdigitation

The compile replace algorithm of Xerox finite state tool conflates the three morphemes into one form giving the stem of the verb. While derivation involving only stem interdigitation can be handled using this regular expression, internal change seems to pose the same problem. One form of verb derivational processes involving internal change is the reduplication, *säbabäräw*. The regular expression for generation of reduplicated stem is given in Figure 2,

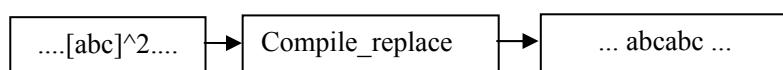


Figure 2 Regular expression for reduplication

In *säbabärä*, the second radical is duplicated and the copy is geminated. This in turn requires the application of two regular expressions, stem interdigitation and reduplication. One may be tempted to formulate a nested regular expression similar to the following,

- (16) -^sbr .m>. CV[CV]^2C <m. [e a e]^] .o. Compile-Replace >> -^sbr .m>. CVCVCV<m. [e a e]^] .o. Compile-Replace >> -säbabär-

Unfortunately, such nested formulation of regular expression is not possible under the current implementation. Moreover, the replica is not an exact match of the original which makes the application of the mechanism proposed for reduplication difficult. Therefore, a separate template form for reduplication has been defined.

Säbbärä	sbr+CVCXVC+ä+Active+Perf+Past+Sing+3rd+Masc
yäsäbr	3rd+Masc+Sing+Active+sbr+CVCC+ä+Verb+Imperf+3P+Masc+Sing
assäbbärä	Causative+sbr+CVCXVC+ä+Verb+Perf+3 rd +Masc+Sing
tesäbbärä	Passive+sbr+CVCXVC+ä+Verb+Perf+3rd+Masc+Sing
säbabärä	Active+sbr+CVCVXXVC+eae+Verb+Perf+ITERT+3P+Masc + Sing

Table 6. Template forms for reduplicative derivations

The output of the above analysis would then be a sequence of morpheme and morphosyntactic features as shown in Table 6. However, since higher level of processing, such as parsing or generation, require more detailed lexical information, the output should be restructured and augmented with more detailed syntactico-semantic information, e.g. argument structure.

4.1 CAT2 Lexical entries

An attempt has also been made to model the Amharic verb morphology using the string unification facility of the CAT2 morphological component. (17) shows a partial specification of a derivational rule for perfect-active form of a tri-radical verb stem.

- (17) Cons={h, l, m, s, r,}
{stem=C1+V+C2+C2+V+C3, voice = act, aspect=perf, tense=past,}.[{lex=C1+C2+C3, C1=Cons, C2=Cons,C3=Cons, V=ä,}].

Along with lexical entry shown in Figure 3, it is possible to model a simple derivation of *säbbärä* from its root form *sbr*.

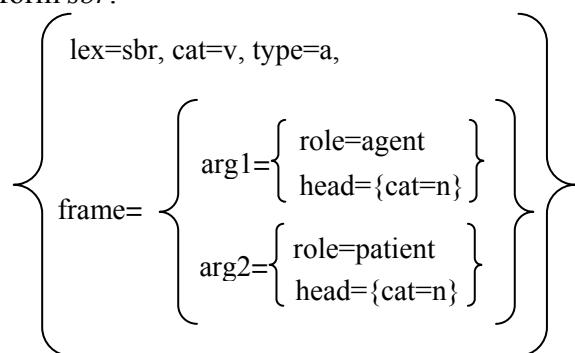


Figure 3 Example lexical entry in CAT2

This approach has some important advantages. First, it provides a sophisticated means of specifying lexical entries using features and unification operation. Second, it is possible to order the rules which may be useful in determining the right order in which the different affixial elements combine with the root form. It also allows modifying the argument structure of a lexeme in some derivational processes. However, run-time unification of the characters and patterns has been found to be a very slow process. Furthermore, some alternation rules are very difficult to model using this mechanism. Hence, this approach has been abandoned, and instead a strategy is now being devised for integrating the two components. One possible strategy which may be envisaged in this connection is that the Xerox tool will be used to analyse word forms and extract all the morphosyntactic features derivable from the surface

form whereas CAT2 can be used to complete the morphological descriptions by augmenting the input with more detailed lexical information, e.g. subcategory and selectional restriction.

5 Conclusion

This paper addressed three related issues: classification of Amharic verbs, aspects of lexical entries for lexical transfer and implementation of morphological analyser. There are a number of proposed frameworks on ways of classifying Amharic verbs. While some provide a detailed classification scheme making lexical specification relatively redundant while others postulate abstract generalizations requiring relatively complex implementation strategy. The strategy adopted in this work is to restrict the process of abstraction only to those frequently occurring phenomena and classes of verbs while introducing some diversity in order to account for some idiosyncratic properties based on some practical consideration. The form of lexical entries is another question raised in this paper. The complexity of Amharic morphology poses difficulty in deciding on the form of the lexeme on which lexical transfer rules apply. Although it is difficult to come up with a proposal, which is applicable to all situations, examination of the different derivational processes shows that a lexeme consisting of root consonants seems a good candidate for lexical transfer. Finally, implementations of Amharic morphological analysis using Xerox finite state tools shows that most of the morphological phenomena can be handled using the finite state operations. However, there are some derivational processes which involve simultaneous application of both stem interdigitation and reduplication operations that can not be accommodated by the current system. This requires stipulation of additional template forms for the derived verbs.

References

- Abiyot Bayou (2000), Developing automatic word parser for Amharic verbs and their derivation, Master Thesis, Addis Ababa University.
- Baye Yimam. (1999) Roots reductions and extention in Amharic. *Ethiopian Journal of Language Studies*. no 9, p. 56-88.
- Bender, M.L and Hailu Fulas (1978). *Amharic Verb Morphology*. Michigan State University.
- Dawkins, C.H (1969), *The Fundamentals of Amharic*, Addis Ababa, Sudan interior mission.
- John J. McCarthy(1985), *Formal Problems in Semitic Phonology and Morphology*, New York
- Kenneth R. Beesley, (1996), Arabic finite-state morphological analysis and generation. In *COLING '96*, vol 1, pages 89-94.
- Leslau, Wolf (1995) *Reference Grammar of Amharic*, Otto Harrassowitz, Wiesbaden.
- Martin Kay (1987), Nonconcatenative finite-state morphology. *EACL '87*, pages 2-10.
- Mesfin Getachew (2001), Automatic part of speech tagging for Amharic language: An experiment using stochastic HMM, Master Thesis, Addis Ababa University.
- Nega Alemayehu (1999) Development of stemming algorithm for Amharic text retrieval, PhD Thesis, University of Sheffield.
- Streiter, Oliver (1996) Linguistic modeling for Multilingual Machine Translation, PhD Thesis, University of Saarland, Shaker Verlag.

SignWriting and SWML: Paving the Way to Sign Language Processing

Antônio Carlos da Rocha Costa and Graçaliz Pereira Dimuro
Escola de Informática, Universidade Católica de Pelotas
96.010-000 Pelotas, RS, Brazil
{rocha,liz}@atlas.ucpel.tche.br

Mots-clefs – Keywords

Langues de Signes, SIGNWRITING, Traitement Automatique de Langues de Signes, SWML
Sign languages, SIGNWRITING, Sign Language Processing, SWML.

Résumé - Abstract

Le système SIGNWRITING est un système pratique pour l'écriture de la langue des signes. Il est composé d'un ensemble intuitif de symboles graphique-schématiques, et de règles pour les combiner dans des représentations de signes. Le langage SWML est un langage basé sur XML pour représenter des textes en langue des signes écrits en SIGNWRITING, de façon indépendante des applications et des plateformes des ordinateurs. Ainsi, des textes écrits en langue des signes, représentés en SIGNWRITING et codés en SWML, peuvent être pris comme entrée par - et aussi obtenus comme sortie de - toute sorte de programme appliquant toute sorte de technique de traitement automatique des langues naturelles (stockage, récupération, analyse, génération, traduction, vérification orthographique, animation, automatisation de dictionnaires, etc.). Ceci ouvre l'ensemble du domaine du traitement automatique des langues naturelles aux langues des signes des sourds. L'article présente les éléments de base d'une telle approche au *traitement automatique des langues des signes*.

The SIGNWRITING system is a practical writing system for deaf sign languages, composed of a set of intuitive graphical-schematic symbols and simple rules for combining them to represent signs. SWML is an XML-based language for encoding sign language texts, written in SIGNWRITING, in an application and computer platform independent way. Thus, sign language texts, written in SIGNWRITING and encoded in SWML, can be entered as input to - and also got as output from - any kind of computer program applying any kind of language and document processing technique (storage and retrieval, analysis and generation, translation, spell-checking, search, animation, dictionary automation, etc.). This opens the whole area of text-based natural language processing and computational linguistics of written texts to the deaf sign languages. The paper presents the basic elements of such approach to *sign language processing*.

1 Introduction

Since the pioneer work of Stokoe (Maher, 1996), deaf sign languages have long been recognized as true natural languages, not as artificial codes. In the same way, deaf culture has been acknowledged as a true minority culture, developed by deaf people when they socially organize themselves within the surrounding hearing society in which they live (Cuxac, 1990).

However important writing systems can be for the consolidation of a culture, deaf people have never developed a practical writing system for sign languages, in spite of such interesting early efforts as Roch-Ambroise Bébian's (Lane, Philip, 1984). Even nowadays the value of writing systems for sign languages find themselves having to be "proven" useful, to be able to gain space in the education of deaf children (Rosenberg, 1999).

Notwithstanding that, deaf educators and individual participants of deaf communities, as well as sign language linguists, have been proposing well-founded notations for deaf sign languages, such as the HAMNOSYS and the SIGNWRITING systems.

The HAMNOSYS system¹ is a scientific notation system, specially designed to be used by linguists in their detailed analytical representation of signs and sign phrases. On the other hand, the SIGNWRITING system² is a practical writing system, composed of a set of intuitive graphical-schematic symbols and of simple rules for combining such symbols to represent signs.

Although it can surely be used in linguistic analytic tasks, the SIGNWRITING system is essentially designed to be used by common (deaf) people, in their daily life. It is conceived to be used in writing sign languages for the same purposes hearing people commonly use written oral languages: taking notes, writing letters, reading books and newspapers, learning at school, making contracts, etc.

This places the SIGNWRITING system in a privileged position to be taken as the preferred writing system for sign language and sign document processing systems, as such systems can thus be put into real practical use by common (deaf) people.

Given the graphical-schematic nature of the SIGNWRITING system, an appropriate encoding of its symbols is necessary, in order to allow the computer storage and processing of sign language document files, as well as the use of written sign languages in interactive control components of computer program interfaces.

That is the purpose of SWML (SIGNWRITING MARKUP LANGUAGE³), an XML-based language that we are developing to allow the computer-platform independent representation of sign language texts written in SIGNWRITING and to allow, thus, the interoperability of SIGNWRITING-based sign language processing systems.

In the following, Section 2 gives an overview of the SIGNWRITING system. Section 3 firstly reviews XML and its role as a meta-language providing for computer systems interoperability. Then, it briefly explains the current version of SWML, the role SWML can play for future sign language processing systems, and the relation it has to the SW-EDIT editor that we are developing for the creation of SIGNWRITING texts and dictionaries. Section 4 pictures the overall scenario of SIGNWRITING-based sign language processing, as envisioned by the approach proposed here. Section 5 brings the Conclusion.

¹<http://www.sign-lang.uni-hamburg.de/Projects/HamNoSys.html>

²<http://www.signwriting.org>

³<http://swml.ucpel.tche.br>

2 The SignWriting System

2.1 Conceptual foundations

Valerie Sutton, the inventor of the SIGNWRITING system, took the stance that, from a practical and intuitive point of view, sign language notation should be visually driven and graphically displayed. Such stance came from her previous experience with the development of a writing system for choreographic movements, the DanceWriting system (Sutton, 1973)⁴.

Sign language notation was, thus, conceived as just another case of *movement writing*, so that the same principles of DANCEWRITING could be applied, and the SIGNWRITING system came up as a *visual notation* for writing sign languages (Sutton, 1999).

Sure, the system was construed to tackle phonetic aspects of sign languages, as they are usually identified by the mainstream of sign language linguistics, e.g., (Valli, Lucas, 1995): hand configurations, hand and finger movements, locations, face expressions, contacts, segmentation, etc. That was necessary because the *visual* aspects of sign languages are precisely what is specific to their linguistic features at the phonetic level (Martin, 2001).

However, in its conceptual foundation, the system was kept as a *movement writing* system, and that is exactly what makes it intuitive and usable for common people, not specially trained in linguistics. Also, that is what makes the SIGNWRITING system neutral with respect to the alternative linguistic frameworks, and thus compatible with otherwise linguistically incompatible theories.

For instance, the SIGNWRITING system is neutral with respect to the various ways to analyze *timing aspects* (sequentiality, simultaneity) in sign language phonology (Valli, Lucas, 1995), and thus is neutral with respect to the *movement-hold segmentation* versus *single segmentation* debate (Uyechi, 1996). It seems to be highly compatible, e.g., with the *visual phonological* approach introduced by Linda Uyechi in (Uyechi, 1996), which was developed well after SIGNWRITING was invented.

2.2 The Graphical Notation

There are various groups of graphical symbols in the SIGNWRITING system, each corresponding to some important (phonetic) aspect of sign languages. The system is permanently evolving, aggregating new elements as they are needed. The two main versions are the SSS-1995 symbol set and the SSS-2002 symbol set.

Figures 1 and 2, below, illustrate the symbols of the SIGNWRITING system, as in the SSS-1995 symbol set. Figure 1 shows the way the system represents basic handshapes. Figure 2 shows the modifications the basic handshape symbols of group 1 may be submitted to, in order to represent different hand orientations and finger configurations⁵. The sample signs are in ASL (American Sign Language).

⁴<http://www.dancewriting.org>

⁵The first three columns correspond to the hand pointing upwards (hand extending in a vertical plane). The following three columns, to the hand point forwards (hand extending in an horizontal plane). Each of the three columns of each plane correspond to the orientation of the palm: facing the signer, facing to the left of the signer, and facing away from the signer.

Figure 3 shows how a text written in SIGNWRITING looks like. It is an extract of an ASL text about ASL grammar, written by Karen van Hoek (Hoek, 1995), and made available free with the SIGNWRITER program. The text is formatted vertically, the preferred orientation of sign language texts for most deafs.

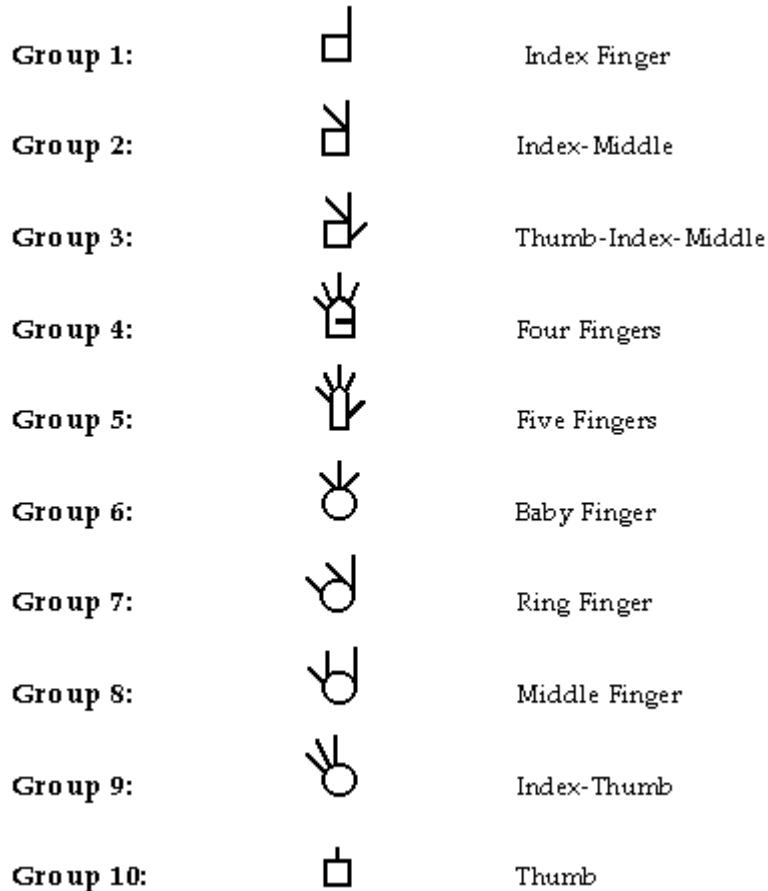


Figure 1: The ten basic handshapes.

3 SignWriting Markup Language

3.1 XML and the Interoperability of Computer Systems

The development of the Internet furthered the need for the interoperability of on-line systems, and XML is the solution proposed by the WORLD WIDE WEB CONSORTIUM (W3C) to such problem⁶. XML is a meta-language allowing the definition of platform- and application-independent languages, dedicated to the storage and processing of information on the Web.

The flexible set of rules incorporated in XML, and the wide availability of both free and commercial software (parsers, checkers, validators, etc.) supporting it, as well as the strong commitment to the language by the main computer manufacturers and software vendors, turned XML

⁶<http://www.w3.org/XML>

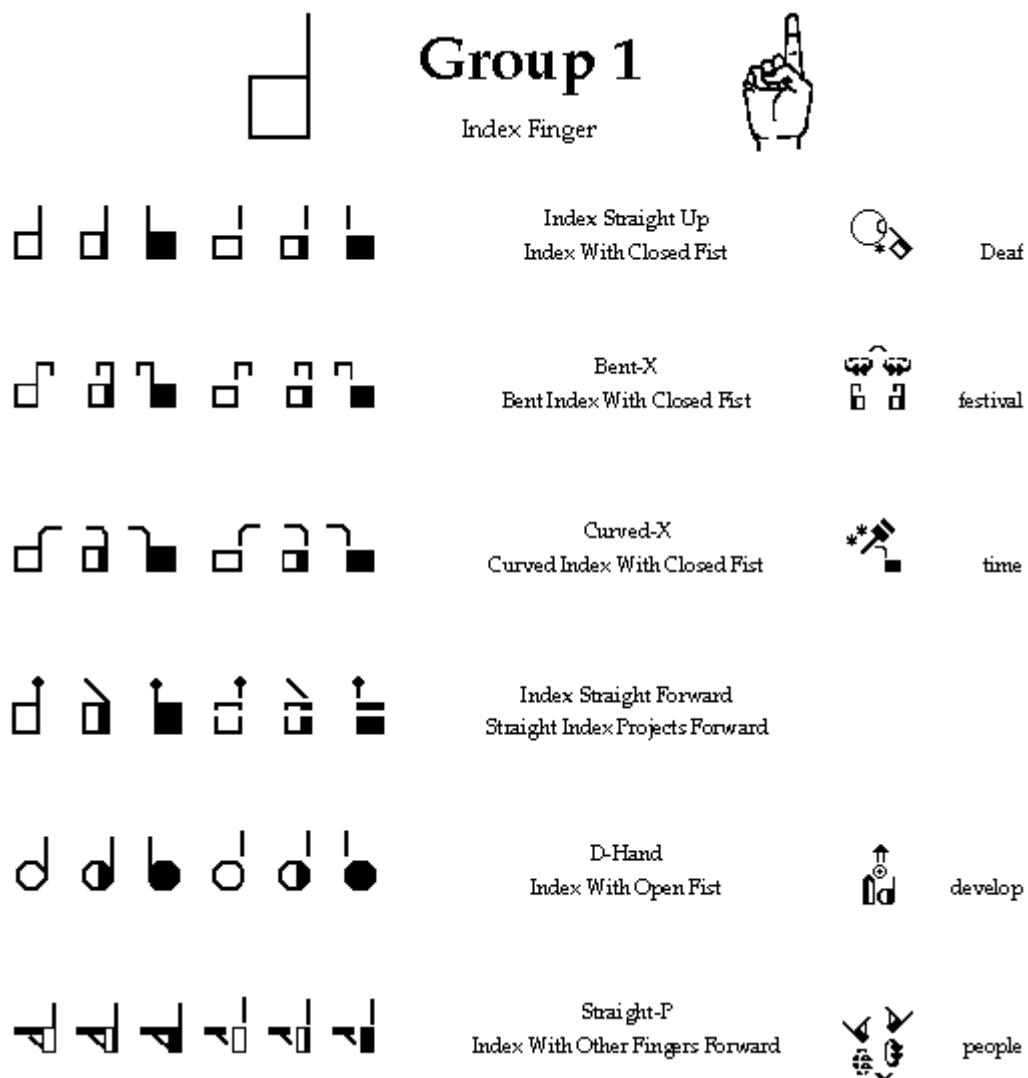


Figure 2: The various modifications of the Index handshape (10).

into the favorite interoperability tool in every software development initiative concerned with that matter.

As it is easy to envision the wide range of applicability of SIGNWRITING on the Internet (email messages, document databases, on-line dictionaries, webpages, chats, etc.), the need of an XML-based format to represent SIGNWRITING files can also be easily understood. The SWML format, explained below, attempts to fulfill such need (Costa, Dimuro, 2001).

3.2 SWML

The SIGNWRITING MARKUP LANGUAGE (SWML) is an XML-based language that is being developed to allow the interoperability of SignWriting-based sign language processing systems.

The current version of SWML is version 1.0, defined by the XML Schema available at <http://swml.ucpel.tche.br/schemas/swml/2003/05/swml.xsd>. Its main features are the following:

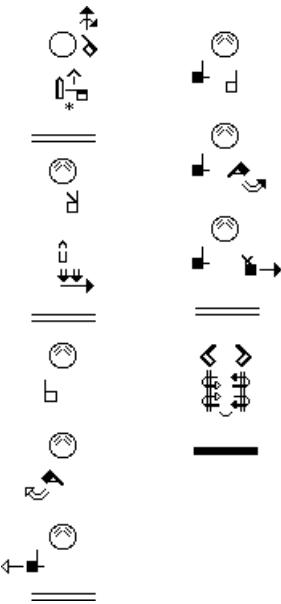


Figure 3: An extract of the *ASL Grammar Lessons*, by Karen van Hoek, written in ASL.

- ◊ SWML can represent both SIGNWRITING texts and dictionaries, as they are generated by either the SIGNWRITER and the SW-EDIT programs.
- ◊ For every sign in the text or dictionary, there is a `<sign_box>` comprising the set of `<symbol>`s that together represent the sign.
- ◊ For every `<symbol>` in a `<sign_box>`, a "number" attribute identifies the `<shape>` of the symbol, and attributes "x" and "y" its coordinates within the `<sign_box>`.
- ◊ Besides, for every symbol, a set of attributes ("variation", "fill" and "rotation") identify the `<transformation>`s to which the symbol was subjected when included in the sign.
- ◊ Two optional attributes, "category" and "group" serve specifically the SSS-2002 symbol set.
- ◊ To support the editing features of the SW-EDIT program, which is a full-fledge, multi-plataform, GUI-based editor, the SWML format defines elements like multi-page documents, page format attributes, inclusion of images, symbol colors, etc.
- ◊ The final result is that the various features (both textual and linguistic) of any sign language text can be extracted from its representation in SWML, thus making the format serve the various purposes of the text and language processing techniques that can be applied to the texts it may represent.

Figure 4 shows the sign for BRAZIL in the Brazilian Sign Language (LIBRAS). The SWML file that encodes such sign is as follows (comments were added afterwards, to ease its reading):



Figure 4: The sign for BRAZIL in Brazilian Sign Language (LIBRAS), written with the SIGNWRITER program, using the SSS-1995 symbol set.

```

<?xml version="1.0" ?>
<swml version="1.0-d2" symbolset="SSS-1995">
    <generator>
        <name>SignWriter</name>
        <version>4.3</version>
    </generator>
    <sw_text>
        <sw_text_defaults>
            <sign_boxes>
                <unit> pt </unit>
                <height> 60 </height>
            </sign_boxes>
            <text_boxes>
                <box_type> graphic_box </box_type>
                <unit> pt </unit>
                <height> 60 </height>
            </text_boxes>
        </sw_text_defaults>
        <new_line/>
        <sign_box>
            <!-- the B hand -->
            <symbol x="8" y="13">
                <shape number="21" fill="1" variation="1" />
                <transform flop="0" rotation="0" />
            </symbol>
            <symbol x="7" y="25">
                <!-- the movement -->
                <shape number="108" fill="0" variation="1" />
                <transform flop="1" rotation="4" />
            </symbol>
        </sign_box>
    </sw_text>
</swml>
```

3.3 SignWriting, UNICODE and ISO Codes for Sign Languages

The ISO Registration Authority has approved in February 2000, as an addition to the ISO 639-2 Standard, the alpha-3 code `sgn` to designate deaf sign languages, with country names added to identify their nationality. Thus, for instance, `sgn-FR` is the code for the French Sign Language, and `sgn-BR` is the code for LIBRAS, the Brazilian Sign Language.

On the other hand, it seems that SignWriting puts some challenges to the UNICODE philosophy of encoding scripts. In particular, the bi-dimensionality of the combinations of SignWriting symbols challenges the linear structure common to most scripts. But if this aspect is solved, the SignWriting symbol set can easily be integrated into UNICODE, and readily put into use in SWML under such encoding.

4 SignWriting-based Sign Language Processing

We use the term *sign language processing* to denote the application of methods and techniques of *natural language processing* and *computational linguistics* to deaf sign languages.

Such methods were originally developed to process oral languages and were, since the beginning, strongly connected to - and even dependent on - methods and techniques of processing oral sentences and discourses presented in *written form*. That was a natural start, given (1) the easiness with which oral (Western European) languages could be represented in computer systems, with the Roman alphabet embedded in the ASCII code, and (2) the socially determined dominance of oral languages.

The extension of that work to non-Western European languages posed (and still poses) interesting technical problems, but has not changed the conceptual foundation of the area, because it still targets only oral languages.

The Gesture Workshop series (Harling, Edwards, 1997; Wachsmuth, Fröhlich, 1998; Braffort et al., 1999) is one of the forums where an alternative goal for natural language processing has shown up, namely, to consider the problem of processing gestures and sign languages.

That work started by dealing with sign language captured visually, in videos or in real time, which was also a natural start, given the lack of standardized (i.e., universally accepted) written form for sign languages.

Some works presented in those workshops dealt with notations for sign languages (e.g., (Lebourque, Gibet, 1999; Vogler, Metaxas, 1999)) but the notations were either linguistically oriented (e.g., based on the STOKOE (Maher, 1996) system or on HAMNOSYS) or computationally oriented (i.e., modeled after some programming language).

Our approach proposes the processing of sign language texts as they may be originally produced by native signers that have no special training in linguistics, and to tackle the problem of common (deaf) user interaction with computer programs using written signs.

Such kind of work, which may well bring to light interesting problems concerning the foundations of natural language processing methods and techniques, can only come up with the help of concepts and tools similar in style to the SIGNWRITING system and SWML.

To pave the way for such kind of work is that we have engaged in the area of sign language processing using the approach explained in the present paper. We are developing very simple computer programs and tools, such as sign counters, manual part-of-speech taggers and simple semantical lexicons, in order to hint on the conceptual problems that should be tackled in the future, when more sophisticated sign language processing systems and techniques may be conceived and proposed.

5 Conclusion

A SIGNWRITING-based approach to sign language processing is possible. Such approach requires a means to guarantee the interoperability of the sign language processing systems based on it. The SWML file format is one such means.

From the point of view of the common (deaf) computer user, such approach may be highly practical and useful, since SIGNWRITING needs no special linguistic training for its use, requiring only that the user learn how to read and write her sign language in such system.

As the SIGNWRITING system was created to be a writing system for daily use, the approach to sign language processing proposed here seems to be in accordance with the system's original intention.

Basic computer programs for processing written sign languages should be developed, to take advantage of texts written with the already existing sign language editors, the SIGNWRITER and the SW-EDIT programs.

As the set of such programs evolve, and users effectively trained in reading and writing sign languages with SIGNWRITING progressively produce growing amounts of sign language texts, and also progressively feedback their experiences in interacting with computers using written sign languages, the stock of sign language processing problems will grow, and assessment of the validity of currently available natural language processing methods and techniques, when applied to sign languages, will be possible.

Sign language processing, besides suffering all the difficulties common to all minority languages, brings a shift in language modality, from the oral-auditive to the gestural-visual modality, that seems to promise interesting new problems for computational linguists.

Acknowledgment

The authors would like to thank the friendship and invaluable continuous support from Valerie Sutton. The research has financial support from CNPq and FAPERGS.

References

- Braffort, A. et al. (eds.) (1999) *Gesture-Based Communication in Human-Computer Interaction*. Berlin: Springer-Verlag. (LNAI 1739 - Proc. Intl. Gesture Workshop, GW'99. Gif-sur-Yvette, France, March 1999.)
- Costa, A. C. R. & Dimuro, G. P. (2001) Supporting Deaf Sign Languages on the Web. *The SignWriting Journal*, v.1, n.0, July 2001. (Available at <http://sw-journal.ucpel.tche.br>. Short version as Poster Paper in the WWW10 Conference CD-ROM, Hawaii, 2001).
- Costa, A. C. R. & Dimuro, G. P. (2002) SignWriting-based Sign Language Processing. In: Wachsmuth, I. and Sowa, T. *Gesture and Sign Language in Human-Computer Interaction*. Berlin: Springer-Verlag.

- Cuxac, C. (1990) *Le Pouvoir des Signes*. In: Sourds et Citoyens. Paris: Institut National de Jeunes Sourds de Paris.
- Harling, P. & Edwards, A. (eds.) (1997) *Progress in Gestural Interaction*. London: Springer-Verlag. (Proc. Gesture Workshop, GW'96. University of York, March, 1996.)
- Hoek, K. v. (1995) *ASL Grammar Lessons*. Written in ASL, with glosses in English. (Published in a SIGNWRITING file that goes with the SIGNWRITER shareware program distribution (Sutton, Gleaves, 1995)).
- Lane, H. & Philip, F. (1984) *The Deaf Experience - Classics in Language and Education*. Cambridge: Harvard University Press.
- Lebourque, T. & Gibet, S. (1999) *A Complete System for the Specification and the Generation of Sign Language Gestures*. In: (Bräffort et al., 1999), p. 227-238.
- Maher, J. (1996) *Seeing Language in Sign - The work of William C. Stokoe*. Washington: Gallaudet University Press.
- Martin, J. (2001) *A Linguistic Comparison - Two Notation Systems for Signed Language: Stokoe Notation and Sutton SignWriting*. Electronic paper, delivered at the SignWriting website: <http://www.signwriting.org/forums/ling008.html>.
- Rosenberg, A. (1999) *Writing Signed Languages - In Support of Adopting an ASL Writing System*. Kansas: Dept. of Linguistics, Univ. of Kansas. (Master's Degree Thesis, available online at <http://www.signwriting.org/forums/rese010.html>).
- Sutton, V. (1999) *Lessons in SignWriting - Textbook and Workbook*. La Jolla: Deaf Action Committee for SignWriting. (2nd ed.)
- Sutton, V. & Gleaves, R. (1995) *SignWriter - The world's first sign language processor*. La Jolla: Deaf Action Committee for SignWriting.
- Sutton, V. (1973) *Sutton Movement Shorthand: a Quick Visual Easy-to-Learn Method of Recording Dance Movement - Book One: The Classical Ballet Key*. Irvine: The Movement Shorthand Society.
- Valli, C. & Lucas, C. (1995) *Linguistics of American Sign Language - an Introduction*. Washington: Gallaudet University Press. (2nd. ed.)
- Uyechi, L. (1996) *The Geometry of Visual Phonology*. Stanford: CSLI Publications. (Dissertations in Linguistics Series).
- Vogler, C. & Metaxas, D. (1999) *Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes*. In: (Bräffort et al., 1999), p.211-224.
- Wachsmuth, I. & Fröhlich, M. (eds.) (1998) *Gesture and Sign Language in Human-Computer Interaction*. Berlin: Springer-Verlag. (LNCS 1371 - Proc. Intl. Gesture Workshop, GW'97. Bielefeld, Germany, September 1997.)

Automatic thesaurus generation for minority languages: an Irish example

Kevin P. Scannell

Department of Mathematics and Computer Science

Saint Louis University

St. Louis, Missouri, USA

`scannell@slu.edu`

Mots-clefs – Keywords

Génération automatique de thésaurus, Irlandais
Automatic thesaurus generation, Irish language

Résumé - Abstract

Nous présentons des techniques pour la génération automatique d'un thésaurus irlandais monolingue. Ces résultats ont été réalisés en dépit des ressources limitées et, comme le plupart des autres langues minoritaires, de l'absence d'outils pour le traitement de langage naturel.

Techniques are presented for the automatic construction of a monolingual Irish language thesaurus. Our results were obtained despite limited resources, including, as is the case for most other minority languages, the lack of sophisticated software tools for natural language processing.

1 Project Description

The goal of this project, taken broadly, is to provide a full suite of Irish language software tools, on par in quality with what is available in English, for everyday use by speakers of Irish. The portion of this work described in the present paper may be of some interest to researchers in computational linguistics, since some of the software that I have developed may be useful in broader contexts and is possibly portable to minority languages other than Irish. I will emphasize the practical versus the theoretical in what follows; such an emphasis is especially important in light of the precarious position of Irish as a spoken language. Given the constant pressure from English (particularly in technical domains) I believe it is essential to focus on producing software that delivers some *immediate benefit* to Irish speakers. The hope, of course, is that providing high-quality Irish software will strengthen the language by reducing (by one) the number of domains in which one is forced to use English. From a sociolinguistic perspective, the technical sphere represents a key battleground in the fight to halt or reverse language shift, particularly in light of Ireland's swiftly developing reliance on technology and the common negative associations of the language with a (real or imagined) backward, rural past.

More specifically, this paper will focus on the development of a hypertext, monolingual Irish thesaurus. In §2-§4 I will provide a detailed description of how the thesaurus was generated, with the hope that the overall process (or indeed some of the specific tools) might be applicable to other minority languages. If nothing else, it should serve as a case study of what can be achieved in this area with severely limited resources.

1.1 Thesauri and automatic thesaurus generation

Roget's English language thesaurus, first published in 1852, is the exemplar of what we will call a *classical thesaurus*: a print or electronic database of quasi-synonyms used most often by writers who are looking for a broad choice of potential synonyms to fit a given context. The basic structure of classical thesauri has remained essentially unchanged over the years; we expect Roget would easily recognize the kernel of his handiwork in the latest editions, despite their abandonment of his original classification scheme for a more convenient alphabetical arrangement, e.g. (Laird, 1999). Classical thesauri usually offer broad coverage of the lexicon and are potentially quite useful tools for the preservation of the rich linguistic heritage of endangered languages like Irish. People with a limited command of the language (as acquired, say, in the national schools) are able to use a thesaurus to expand their vocabulary and improve their writing.

It is convenient to distinguish classical thesauri from *electronic thesauri* in the modern sense: software components used in many document retrieval or indexing systems, usually for the selection of a preferred form of a given search term. The underlying data in classical and electronic thesauri are quite similar (raw lists of terms organized according to some kind of semantic hierarchy) and our goal in this project is to generate a common database of semantic relationships in Irish from which, initially, a classical thesaurus can be generated, but with the flexibility that in the future more sophisticated information retrieval tools can be developed.

There is a rich literature covering techniques for automatic thesaurus generation, but most of the work has been restricted to global languages. The best references for the elements of thesaurus construction are (Aitchison & Gilchrist, 1987), (Grefenstette, 1994), and the ANSI/NISO

standard *Guidelines for the Construction, Format, and Management of Monolingual Thesauri* (ANSI/NISO, 1993). Typical systems parse a large corpus and apply some form of cluster analysis either to measurements of similarity in grammatical context or to raw counts of co-occurrence. In particular, all approaches of which we are aware rely on a sophisticated pre-existing NLP infrastructure (large corpora, parsing tools, etc.), taken for granted in languages like English but not available in Irish or most other minority languages.

If the ultimate goal of automatic thesaurus construction is the deduction of semantic relationships exclusively from free text corpora, systems may be viewed as more or less technically remarkable as their underlying corpora vary from free to highly-structured. According to this measure, our approach is decidedly unremarkable, as the main idea is to exploit existing English language thesauri to deduce the desired semantic relationships in Irish.

1.2 Survey of available resources

What I hope is inspiring about this case study is the fact that the end results have been achieved with virtually no financial resources, no pre-existing software infrastructure, and a limited time commitment¹. As will become clear in a moment, though, any such inspiration must be tempered by the fact that Irish, compared with other minority languages, enjoys an embarrassment of lexicographic riches in machine-readable form. The approach I describe in §2-§4 may therefore not be feasible for the most severely marginalized languages.

A broad survey of Irish language resources on the Internet can be found at the *Gaeilge ar an Ghréasán* site maintained at Sabhal Mór Ostaig². Of special interest are several online newspapers either entirely in Irish³ or devoting special sections to Irish language articles⁴. Highly informal writing and cutting-edge usages can be gleaned from the archives of several online discussion groups⁵, while the recently released CD ROM version of the Bible (Ó Fiannachta, 1981) provides a convenient source of formal literary material⁶. Most useful for lexicographic work are the resources made available by the Irish government⁷ (specifically *An Coiste Téarmaíochta*, who are in charge of coining modern terminology, and *An Gúim*, the government publishing house), and by *Fiontar*, a program at Dublin City University devoted to interdisciplinary studies through the medium of Irish⁸.

Irish speakers also benefit from several outstanding print resources. These include the two standard bilingual dictionaries (Ó Dónaill, 1977) and (de Bhaldraithe, 1959), and a recently published monolingual thesaurus (Ó Doibhlin, 1998). Though on a much smaller scale than the present work, the latter is a finely crafted book, produced (presumably manually) by a fluent speaker and Irish language scholar. I have intentionally not incorporated its contents in the current version of the database, so that it can provide an objective “gold standard” measure of quality of the computer-generated output. Examples are discussed in §4.

While surveying the available corpus material and discussing the limitations on financial re-

¹My primary research areas are in pure mathematics and theoretical physics.

²See <http://www.smo.uhi.ac.uk/gaeilge/gaeilge.html>

³e.g. <http://www.beo.ie/>

⁴e.g. <http://www.ireland.com/gaeilge/teangabeo/>

⁵e.g. <http://listserv.heanet.ie/lists/gaeilge-a.html>

⁶See <http://www.fiosfeasa.com/>

⁷See <http://www.acmhainn.ie/>

⁸See <http://www.dcu.ie/fiontar/further/focloiri.html>

sources I should also note that there are two extent Irish corpora that I have not used; a substantial one developed as part of the European Union PAROLE project (prohibitively expensive at 250 euro) and a somewhat smaller one compiled by Ciarán Ó Duibhín⁹ (free, but for use only on Windows machines).

2 Phase One: Creating a software infrastructure

The first step in the process involved the development of some simple lexicographical database software. Naturally, a great deal of the effort that went into this phase could have been avoided by using an existing package. On the other hand, starting from scratch has made it easier to integrate successive phases with the underlying database, and, where necessary, to tailor things to the specific needs of Irish. A typical record in the database stores a dictionary headword, basic grammatical information (including tags for special inflections), and a list of citations.

Each record also stores, recursively, a list of records in the same format representing alternate forms. Careful handling of these alternates is essential for a language like Irish which had no standardized orthography until the middle of the 20th century (Rannóg an Aistriúcháin, 1962), and for which the standard has not taken root in the hearts of all native speakers. The majority of alternate forms in the current version of the database are either pre-standard or dialect forms, with a sprinkling of modern terminology that has been subsumed or made obsolete (e.g. a word like *glaothán* (“a pager”) that appeared in (Mac Mathúna & Ó Corráin, 1995) almost ten years ago but has been supplanted by *glaoire* in usage and in the recommendations of *An Coiste Téarmaíochta*).

Next, I wrote a program in C++ called *morph-ga* that generates all inflected forms of Irish nouns, adjectives, and verbs when provided with a headword and sufficient grammatical tagging information. This piece of software is the linchpin for everything that follows, in particular providing a useful shortcut that I call “naïve stemming”. Instead of taking the time to write a completely general stemmer, it suffices to implement some basic heuristics for making wild guesses at stems. Suppose, for instance, that a target word *mhantaí* appears in a corpus text. The software recognizes the ending *-aí* as (1) a common plural ending, (2) the comparative ending of an adjective ending in *-ach*, or (3) a rarely used verb ending in the subjunctive (Irish speakers may see other possibilities which must be disposed of as well). Heuristic (1) leads to a conjectural noun stem *mant* which is indeed found in the database, but *morph-ga* correctly generates its plural as *mantanna*, eliminating this case. Heuristics (2) and (3) yield *mantach* and *mantaigh* respectively, and the target word is found as a correct morphological form in each case. Probability says that possibility (2) is surely correct, but contextual markers must be used to verify this for certain.

3 Phase Two: Generating a clean list of words

The goal of this phase was, in short, to fill up the database created in phase one. Most important was the creation of an accurate list of dictionary headwords with complete grammatical information. Of secondary importance were accurate citations to print and electronic texts.

⁹<http://www.smo.uhi.ac.uk/~oduibhin/tobar/>

3.1 Methodology

1. **Extract the core database from a corpus.** I began by assembling a small corpus of electronic material out of the sources noted in §1.2 and wrote shell scripts that hunt for forms not already in the database, sorting by frequency. Later, improved, versions assign “editorial” weights to different texts and count an appearance in, say, the carefully edited *Oll-liosta Téarmaíochta* more heavily than one in the archives of an email discussion group. Naturally one expects that the words at the top of the list are the ones most likely to be spelled correctly; these are run through the stemmer and incorporated into the database (assuming they pass the various checks below).
2. **Add citations from print dictionaries.** A certain amount of checking by hand against print dictionaries has been performed as well, as a way of verifying the accuracy of the words being added to the database, but also as a way of fleshing out the lists of citations which are used in various ways during later phases of the project. In addition to the standard bilingual dictionaries, there are several books of terminology in print (Biology, Home Economics, Geography, etc.) representing the work of *An Coiste Téarmaíochta*. In an afternoon, one can add citations from one of these dictionaries to the database with a single keystroke per entry.
3. **Validate spelling via pattern matching.** Another powerful tool for checking the database is a shell script that uses pattern matching to look for illegal combinations of characters in a raw word list. The current version of this script implements 200 rules, varying from the trivial (only the characters ‘l’, ‘n’, and ‘r’ are doubled in Irish) to the subtle (a string of consonants preceded by a so-called broad vowel – ‘a’, ‘o’, or ‘u’ – is in general not allowed to be followed by a slender vowel – ‘e’ or ‘i’). While there are many exceptions to certain rules, these exceptions can be either verified by hand or further whittled with some addition pattern matching.
4. **Validate spelling via authoritative texts.** The citation information garnered in step two is exploited to look for potential spelling problems as follows. Each source is assigned a weight that measures its “authoritativeness” from the point of view of spelling (thus modern print dictionaries get high values while materials produced before the spelling reform in the 1940’s get extremely low values). Warning flags can be raised when an alternate form has a more authoritative citation than the putatively standard form, or, similarly, if an alternate form has a greater number of citations than the standard form (authoritative or not).

3.2 Results

The data assembled in this phase enabled us to distribute the first full-scale Irish spellchecker, originally packaged for use with Geoff Kuenning’s *International Ispell* and released under the GNU public license¹⁰ in June of 2000. This initial release contained just over 13,000 dictionary headwords and some 171,000 inflected forms. Since then, I have repackaged things for use with the other widely-used spellcheckers in the open source community (`aspell` and `myspell`) and the database has grown to almost 30,000 headwords and 300,000 inflected forms¹¹. Diar-

¹⁰See <http://www.gnu.org/copyleft/gpl.html>

¹¹Available from <http://borel.slu.edu/ispell/>

maid Mac Mathúna has recently repackaged the word lists for use with Microsoft software, maintaining the open source license.

My guess is that the percentage of remaining misspellings (as of February 2003) is probably smaller than for some widely-used English spellcheckers (if so, this would be one of the rare instances in which the minority language tool outstrips the English language tool).

4 Phase Three: Generating the thesaurus

The goal of this phase was to generate a machine-readable thesaurus that can be output, for example, as a high-quality PDF document with hypertext links. Eventually we hope to refine the process described below to have the output compliant with the ANSI/NISO Z39.19 standard (ANSI/NISO, 1993). This will allow the database to be integrated more easily into information retrieval or indexing systems that rely on the standard.

The key labor-saving idea here is the introduction of English translations, allowing us to transfer semantic relationships from existing English language thesauri to Irish. While engaged in this work, I learned of a pilot study done at the University of Limerick that is akin in spirit to our approach (Sutcliffe *et al.*, 1996). They describe a prototype of a multilingual version of WordNet which, essentially, maps words from non-English languages into the existing WordNet hierarchy¹². Modulo the ongoing port of our database to the WordNet format (discussed below), our work provides a full-scale realization of the system envisaged in their paper.

The introduction of English may raise some theoretical worries that we shall address in §4.2.

4.1 Methodology

1. **Assign raw English meanings to headwords.** This was surely the most labor intensive phase, though it was made easier by the resources at www.acmhainn.ie and several other small-to-medium scale English-Irish and Irish-English electronic glossaries produced by amateur language enthusiasts¹³. Where necessary, lists of English meanings were fleshed out by reference to the standard print dictionaries (Ó Dónaill, 1977), (de Bhaldraithe, 1959), (Mac Mathúna & Ó Corráin, 1995), and (Ó Cróinin, 2000).
2. **Resolve ambiguities among English definitions.** Much of this step can be automated via standard word sense disambiguation techniques, though in doing so we relied to a certain extent on the quality of the available Irish-English dictionaries. For instance, one rarely finds a single polysemous English translation for a given headword in (Ó Dónaill, 1977), even when a human reader would surely know the correct resolution. By using a database of polysemous English words and a scheme for resolving them, as provided by a system like WordNet (Fellbaum, 1998), the software can easily decide, for instance, that the word *feileastram* with English translations “iris, flag” refers to a plant and not part of the eye or a kind of banner. When there are not sufficiently many English translations or if the translations are missing from the English database, some human intervention becomes necessary. In reality, instead of doing the sensible thing and using WordNet

¹²The prototype can be found at <http://nlp01.cs.ul.ie/iwn.html>

¹³See <http://www.crannog.ie/focloir.htm> for a notable 14,000 word example.

from the beginning, I developed my own primitive version based on the public domain Roget’s Thesaurus (Roget, 1991)¹⁴. Were I to do it all over, I would surely use WordNet in light of the time savings, improved quality, and standardization its use would represent. I may “port” the resolutions in the database to this format at some future date.

3. **Break word list into semantic equivalence classes.** The idea here is a completely naïve but seems to work well. To first order, we tentatively assign two words to the same equivalence class when they share a resolved English translation. This assignment is given a “confidence parameter” that increases when there are multiple shared translations. More generally, whenever two Irish words have resolved English translations that are (possibly different but) semantically close (as determined by reference to an English thesaurus) the confidence parameter is increased by an amount proportional to the semantic proximity of the English translations (equality naturally providing the largest increase). The terminology “equivalence class” is perhaps deceiving here, since transitivity of the equivalence relation fails badly. Were one to take the transitive closure by, say, further increasing the confidence parameter between two words if there is a chain of equivalences joining them, essentially unrelated words would end up marked as equivalent. For example, one might guess incorrectly that *gearchúiseach* (“shrewd”) is related to *garg* (“pungent”) since the polysemous Irish word *géar* shares each of these English senses. Though we have no *a priori* method for disambiguation of Irish words, there is clearly potential for some bootstrapping here. The thesaurus generated at this step (without transitivity) implicitly picks out the different senses of a word like *géar*; one could then implement transitivity as suggested above when there exists a chain of equivalences between *disambiguated* Irish words.
4. **Generate the hypertext thesaurus.** This step converts the internal database of equivalence classes into a human-readable format (namely, hyperlinked PDF). Representative nouns were selected for about 1000 basic categories, similar to the classical Roget’s thesaurus in English. This was done automatically, through a combination of criteria involving (1) the frequency of appearance of the representative word in the corpus, (2) a measure of its centrality in the equivalence class, and (3) its lack of ambiguity. The current PDF version displays the thesaurus in alphabetical order, each entry being followed by one or more hypertext links to the representative word(s) under which it appears. Preliminary versions are available for free download¹⁵.

4.2 Results

As noted above, the use of English translations ought to raise some concerns about this phase in the process. The potential imposition of English language categorizations into a monolingual Irish thesaurus will surely raise some Whorfian hackles. This may be perceived as particularly dangerous ground for an endangered language; Irish readers will be reminded of Tomás Ó Rathaille’s famous characterization of the then moribund Manx language as “English disguised in Manx vocabulary” (Ó Rathaille, 1932). Unfortunately, this is the sort of corner into which one is forced when working with a minority language lacking any substantial monolingual lexicographic material.

¹⁴ Available from <http://www.promo.net/pg/>

¹⁵ <http://borel.slu.edu/teasaras/>

Our theoretical defense rests first on the *coarse granularity* of thesauri, that is, the semantic fuzziness inherent in a long list of quasi-synonyms. Take the canonically “untranslatable” Irish word *dúchas* in its most abstract sense of “heritage, patrimony”. Though these English translations are a poor reflection of the depth of meaning in the Irish word, they are also given as translations of its nearest Irish synonym, *oidhreacht* (also meaning “inheritance” in the concrete sense). Thus, since we are not concerned with razor-sharp precision but only that these Irish words end up near each other in the thesaurus, the algorithm above suffices.

A relativist criticism is also weakened somewhat when leveled against the English-Irish language pair which has seen, for better or for worse, several centuries of heavy (mostly unilateral) lexical borrowing. One would probably need more care in trying our approach with, say, Hopi or Dyirbal.

Fundamentally, though, our strongest argument is the *a posteriori* one provided by the quality of the finished product. As noted in the introduction, an objective measure of quality can be obtained by comparing selected portions of the output with a “gold standard” thesaurus (Grefenstette, 1994) for which we use the *Foclóir Analógach* (Ó Doibhlin, 1998).

Here, for example, is the entry from (Ó Doibhlin, 1998) under the headword *anachain* (“misfortune, adversity, calamity”). It is divided into two halves, the first listing 29 general varieties of adversity and the second listing 27 more specific calamitous occurrences.

Cineálacha: Angar. Mí-ádh. Míchinniúint. Mífhortún. Míshéan. Léan. Doilfós. Tubaiste. Donas. Ainnise. Léirsrios. Báine. Buaireamh. Anacair. Anbhroid. Anó. Imní. Duainéis. Crá. Clipeadh. Bearrán. Leatrom. Lionn dubh. Díomá. Drochmhisneach. Diomú. Beaguchtach. Éadóchas. Buille fill.

Saintarlú: Tubaiste. Tionóisc, taisme. Turraing. Stoirm. Anfa. Spéirling. Tíofún. Gorta. Plá. Cogadh. Tuile. Dóiteán. Ár. Éirleach. Longbhriseadh. Coscairt. Scríos. Slad. Creach. Dochar, damáiste. Turnamh. “Tonnbhriseadh an tseanaghnáthaimh”. Teip. Titim. Briseadh.

Now, the corresponding entry from our thesaurus (138 words/phrases). The order of words is dictated by the software, which lumps together words deemed to be semantically close according the measures discussed above.

tubaiste, drámh, donacht, uisce an cheatha, lomadh an Luain, cinniúint, caill, dochma, teann, crácamas, dua, anró, easonóir, síleáil, tónáiste, tiortáil, cruanan, splíontaíocht, mífhortún, cora crua an tsaoil, droch-chor, ciotaí, treampán, callshaoth, anchaoi, aimpléis, drochrath, caduaic, seacht gcúraimí an tsléibhe, seacht gcúraimí an tsaoil, iomard, umar na haimlaise, duibheagán an éadóchais, anrath, mírath, pláinéad, saol léanmhar, saol crua, saol anróiteach, drochshaol, mí-ádh, dochraide, deacair, cacht, bráca, toirmeasc, míchonách, duais, deacracht, cruachás, smál, buairt, buaireamh, margadh éagórach, anachain, trioblóid, diachair, donas, maírg, bris, angar, dobrón, cránas, léan, dothairne, dólás, doilfós, leatrom, doghrainn, tinneas, guais, géibheann, éigeantas, éigean, duainéis, dola, dócúl, dochar, broid, anó, anbhroid, anacair, sníomh, imní, triail, cros, céasadh, cath, sciúrsáil, crá, drochanáil, pionós, plá, sciúirse, imirt, gearradh, tuisle, tapaigean, míthapa, tionóisc, óspairt, timpiste, taisme, púir, ochlán, liach, dursan, cat mara, matalang, turraing, gátar, eirleach, meath, díomua, céim síos, dul ar gcúl, gonic, longbhriseadh, titim, turnamh, treascairt, milleadh, creachadh, faillí, cliseadh, teip, meathlú, meathlaíocht, loiceadh, feall, scríos, raic, díothú, creach, cabhóig, anás, ainriocht, aimhleas.

For reasons of space, we will restrict ourselves to a few simple observations. First, because our underlying English thesaurus tends not to give lists of specific kinds of things, our output leaves out nine of the ten calamities starting at the cognate *stoirm* and ending at *ár* “slaughter” (we picked up *plá* only because of its figurative use as “a scourge”). This seems to be a matter of taste in thesaurus construction versus a linguistic issue.

Leaving out these ten, we hit 28 of 46 ($\approx 60\%$). We missed words in places where Ó Doibhlin seems to stray farther afield from the central meaning of “adversity”: *léirscrios*, *báine* (“destruction”), *clipeadh*, *bearrán* (“teasing”), and the final seven of the first list, all variants of “sorrow” or “despair”. Undoubtedly we should have picked up *míchinniúint* which was in our database but was a bit light on English translations (“ill fate”).

The good news, of course, is the incredible richness of expression found in the expanded list. Even the most fluent speakers, we hope, will discover new idioms (*lomadh an Luain*), unusual secondary meanings (*pláinéad* as “ill luck, planetary influence”), or literary words that have fallen into disuse (*cacht*, *dursan*).

Finally, we do not see any “howlers” in the output (though a poor job of disambiguation of English definitions has led to some embarrassing blunders in other lists).

We emphasize that the example just given is the *unedited output* of the sequence of algorithms described above. A change to the underlying database (say, the addition of a new English definition for an Irish word) automatically propagates itself (sometimes in subtle ways) when we give the command to rebuild the thesaurus from scratch. This enables continuous updating of the thesaurus, and allows end users to make contributions or corrections in a standardized way. Such continuous maintenance is essential for any piece of software, but especially so when the primary goal of the software is the accurate reflection of the various idiosyncrasies of a living language: new terminology, shifting usages, etc.

We believe this kind of collective approach to software development and maintenance will be essential to the future provision of quality software to speakers of minority languages. In concrete terms, this approach is facilitated by releasing our thesaurus and its L^AT_EX sources under the GNU Free Documentation License¹⁶ which says, in short, that everyone has the freedom to copy, modify, or even sell the thesaurus as long as redistributed versions preserve the same freedoms. This kind of license guarantees the widest possible dissemination of the materials we have developed, but more importantly, empowers speakers of minority languages by placing control of these resources directly in their hands, eliminating the generally fruitless reliance on the benevolence of large corporations for the provision of such material.

Acknowledgments

Thanks to Diarmaid Mac Mathúna, Michael Conry, Vincent Morley, and Alastair McKinstry for their interest in the spellchecking project and for helping to spread the news of its availability. Alastair, in particular, deserves credit for developing the first non-trivial Irish spellchecker (circa 1997). I benefitted from enjoyable email exchanges with Andrew Dunbar and Alan Horkan about machine translation and software localization, respectively. Many people deserve credit for producing the electronic texts upon which this work was based, but especially helpful were Caoimhín Ó Donnaíle (who provided some well-edited lexicographic material) and Antain Mac

¹⁶See <http://www.gnu.org/licenses/fdl.html>

Lochlainn and the people at www.acmhainn.ie who have made available much of the work of *An Coiste Téarmaíochta*.

References

- AITCHISON J. & GILCHRIST A. (1987). *Thesaurus construction: a practical manual*. Aslib, London, 2nd edition.
- ANSI/NISO (1993). Z39.19 – 1993 Guidelines for the Construction, Format, and Management of Monolingual Thesauri.
- T. DE BHALDRAITHE, Ed. (1959). *English-Irish Dictionary*. An Gúm, Baile Átha Cliath.
- FELLBAUM C. D. (1998). *WordNet: an electronic lexical database*. MIT Press, Cambridge, Mass.-London.
- GREFENSTETTE G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer Acad. Publ., Dordrecht.
- C. LAIRD, Ed. (1999). *Webster's New World Roget's A-Z Thesaurus*. Macmillan, New York.
- S. MAC MATHÚNA & A. Ó CORRÁIN, Eds. (1995). *Collins Gem Irish Dictionary*. Harper-Collins Publishers, New York.
- B. Ó CRÓININ, Ed. (2000). *Pocket Oxford Irish Dictionary*. Oxford Univ. Press, Oxford.
- Ó DOIBHLIN B. (1998). *Gaoth an Fhocaill*. Coiscéim, Baile Átha Cliath.
- N. Ó DÓNAILL, Ed. (1977). *Foclóir Gaeilge-Béarla*. An Gúm, Baile Átha Cliath.
- P. Ó FIANNACHTA, Ed. (1981). *An Bíobla Naofa*. An Sagart, Maigh Nuad.
- Ó RATHAILLE T. (1932). *Irish dialects past and present*. Institiúid Árd-Léinn, Baile Átha Cliath.
- RANNÓG AN AISTRÍUCHÁIN (1962). *Gramadach na Gaeilge agus Litriú na Gaeilge: An Caighdeán Oifigiúil*. Oifig an tSoláthair, Baile Átha Cliath.
- ROGET P. M. (1991). Project Gutenberg Roget's Thesaurus.
- SUTCLIFFE R. F. E., O'SULLIVAN D., MCCELLIGOTT A. & Ó NÉILL G. (1996). Irish-English mappings in International WordNet: a pilot study. Unpublished.

Disambiguation of case suffixes in Basque

Victor Lascurain (1), Eneko Agirre (1), Mikel Lersundi (1)
Luboš Popelínský (2)

(1) University of the Basque Country, Donostia, Spain
Email: bittor@web.de, eneko@si.ehu.es,
jialeaym@si.ehu.es
(2) Faculty of Informatics, Masaryk University
Botanická 68, CZ-602 00 Brno, Czech Republic
Email: popel@fi.muni.cz

Mots-clefs – Keywords

désambiguation morphologique, désambiguation de la sens du mot, apprentissage inductive
morphological disambiguation, word-sense disambiguation, inductive learning

Résumé - Abstract

Le but de ce projet étais la classification automatique des cas grammaticaux dans la langue Basque. Pour réaliser ça on a appliqué l'apprentissage inductive (les systèmes Tilde et Timbl). On emploie WordNet pour retrouver des mots et les hyperonyma des mots dans un contexte. L'exactitude étais plus haut que 70% pour Tilde et 63% en cas du Timbl.

The goal of this paper is to build a tool for automatic classification of grammatical cases in Basque. To achieve this goal we applied inductive learning techniques, namely systems Tilde and Timbl. We use WordNet for finding synsets and hyperonyms of words in a context. For Tilde we reached accuracy higher than 70% and for Timbl 63%.

1 Introduction

The goal of this paper is to build a tool for the automatic classification of case-suffixes in Basque. Basque is an agglutinative language and its case suffixes are more or less equivalent to prepositions, but they are also used to mark subject and objects of verbs. If we want to disambiguate the relation between the verb and the prepositional phrase it is really important to know the possible interpretations of the Basque case-suffixes, as it is important in other languages to know the possible interpretations of prepositions. This can be considered as a semantic disambiguation task.

All case-suffixes used in Basque need to be analysed. Some of them are more ambiguous than the others as it happens with prepositions in other languages. We have chosen *instrumental* because it is one of the most ambiguous, and more interesting from a disambiguation point of view. To clarify the task, Table 1 shows the possible interpretations of the instrumental case-suffix with examples in Basque and their translation into English.

Table 1: Possible interpretations of the instrumental case-suffix (-z).

	Basque	English
theme	Seguru nago horretaz Matematikaz asko daki	I'm sure of that He's an expert in maths
during-time	Arratsaldez lasai egon nahi dut Gaeuz egin dut	I like to relax of an evening I did it by night
instrument	Autobusez etorri naiz Belarra segaz moztu Euskaraz hitz egin	I have come by bus To cut grass with a scythe To speak in Basque
manner	Animali baten hestea betez egindako haragia Ahots ozen batez	A meat preparation made by filling an animal intestine In a loud voice
cause	Haren aitzakiez nekatuta nago Beldurrez zurbildu Kanpoan lan egitea baztertu zuenez, lan-aukera ederra galdu zuen	Sick of his excuses To turn white of fear In refusing to work abroad, she missed an excellent job opportunity
containing	Edalontzia ardoz beteta dago Txapelaz dagoen gizona Ilez estalia	The glass is full of wine The man with the beret on Cover in hair
matter	Armairua egurrez egina dago	The wardrobe is made of wood

The goal is then to classify each occurrence of the case suffix into one of the possible interpretations, – *theme*, *place*, *instrument*, etc. – taken into account the context. The approach we have used is based on learning from a set of hand-tagged occurrences of the instrumental case suffix, using inductive logic programming (Muggleton, 1992; Muggleton & De Raedt, 1994) (ILP) and instance-based learning (Zavřel & Daelmans, 1998) techniques. In order to test each of the approaches we have used 5-cross validation.

The context of each occurrence is annotated with ambiguous morphological tags. It means that for all context words we know all morphological readings but we do not know the right one. Besides WordNet¹ is used to generalise the words in the context. WordNet is a lexical reference system that organise English nouns, verbs, adjectives and adverbs into synonym sets, each representing one underlying lexical concept.

The structure of this paper is following. In Section 2 we describe the data used for learning. Section 3 contains brief information on WordNet. Experiments with the ILP system Tilde are described in Section 4. In Section 5 we bring results obtained with the instance-based learner Timbl. We conclude with overview of relevant works and with concluding remarks.

¹<http://www.cogsci.princeton.edu/~wn/>

2 Data

The learning database contained 142 correctly classified examples of the target relation. These examples have been extracted from a monolingual Basque dictionary (Sarasola, 1996). Each example is a sentence containing a word inflected in the instrumental case. Each word in the sentence has been ambiguously morphologically tagged.

One example in a raw format shown in Figure 1 represents the sentence “Bazkaz hornitu.”. Each word of the sentence is enclosed in “<>” and followed by a list of possible readings. The first word, “<Bazkaz>” has two possible readings. In each reading the first part is the lemma (“bazka” *pasture* or *grass* in English) followed by a list of tags. We exploit here only the morphological ones. For this word the interesting tags are “IZE” (noun), “DEK” (declined word) and “INS” (instrumental case). The second word (“hornitu”, *feed* in English) has three possible readings. In this case we can see that the lemma is always followed by the tag “ADI” (verb), so this word has only verb readings.

Figure 1: An example of a sentence in a raw format.

```
"<Bazkaz>
  "bazka"  IZE ARR DEK INS MG
  "bazka"  IZE ARR DEK INS NUMS MUGM
"<hornitu>
  "hornitu"  ADI SIN AMM PART ASP BURU  NOTDEK
  "hornitu"  ADI SIN AMM PART DEK ABS MG
  "hornitu"  ADI SIN AMM PART  NOTDEK
"<$.>$"
  PUNT_PUNT
```

The data in this format was further transformed into the form of Prolog facts. Each example consists of three predicates, *position/1* (the word carrying the instrumental case), *leftCtx/1* (the list of words in the left context, in the reverse order, together with their morphological readings), and *rightCtx/1* (the list of words in the right context with their morphological readings). The transformed data can be seen in Figure 2. Each example in the learning set has been man-

Figure 2: Sentence from the database. Prolog format.

```
begin(model(example1)). theme.
leftCtx([]).
rightCtx([word(hornitu,
  [[hornitu,adi,sin,amm,part,asp,buru,notdek],
   [hornitu,adi,sin,amm,part,dek,abs,mg],
   [hornitu,adi,sin,amm,part,notdek]]]]).
position(word(bazkaz,
  [[[bazka,ize,arr,dek,ins,mg,aorg,has_mai,def_hasi,notgelgen],
    [bazka,ize,arr,dek,ins,nums,mugm,aorg,has_mai,def_hasi,
     notgelgen]]]]).
end(model(example1)).
```

ually classified into one of seven different semantic categories. Their frequency is displayed in Table 2.

3 WordNet

The most important information for our task is the meaning of the word present in the relation the case suffix represents, usually a noun and a verb. As it is impossible to list every single word

Table 2: List of semantic categories and corresponding frequencies.

Class:	cause	containing	instrument	manner	matter	theme	time
Number:	5	23	31	41	7	29	6
Frequency:	0.03	0.16	0.21	0.28	0.05	0.20	0.04

pair that can be related to a given preposition or case, some way of generalisation from words to more abstract concepts will be useful. For this crucial task we exploited information from WordNet. The WordNet is a net made of words, or more exactly, synsets. A synset is a named collection of words that share a common meaning. Synsets in the net are related to each other in many ways. Regarding our work the *hyperonymy/hyponymy* relation is the most important. This relation defines a sub net inside the WordNet, which links synsets regarding to a *is-a* relation. This relation enables generalisation from specific words to more general concepts.

4 Learning with Tilde

Inductive logic programming (ILP) (Muggleton, 1992; Muggleton & De Raedt, 1994) is a machine learning technique that learns first order logic descriptions from a set of examples and a given background knowledge (Muggleton, 1992; Muggleton & De Raedt, 1994). We used the *Tilde* system which learns first order logic decision trees (Blockeel & Raedt, 1997).

Good background knowledge expressed in the form of a logic program is crucial for a good performance of any ILP system. We tested several different types of background knowledge predicates. A description of the characteristics of each of them as well as the obtained results are presented in the next paragraphs.

4.1 Morphological predicates

The first set of predicates is composed of only simple morphological predicates. There are two different types of predicate, *exists* and *forall* predicates. The “*exists/I*” predicate checks whether a given morphological tag is present in at least one of the readings of one of the words in the example. The “*forall/I*” predicate checks whether a given morphological tag is present in all the readings of at least one word in the example. The accuracy of this classifier is around 47%. All the results have been obtained by running 5-cross validation.

A second experiment was done with a slightly modified set of predicates. Namely “*exists/2*” and “*forall/2*” predicates were added. They do the same checks as their arity 1 equivalents but for a pair of tags instead for a single one. The accuracy increased to 55%.

4.2 Semantic (WordNet) predicates

In order to increase accuracy semantic information from WordNet has been introduced into the background knowledge. This semantic information was used in both possible ways, either alone, or in combination with the morphological information. The new predicates have the

form $\text{hasSynset}(X)$ and $\text{hasHyperonym}(X)$ till 3rd level up in the hyperonymy hierarchy. The predicate $\text{has_synset}(\text{Synset})$ succeeds if a word, member of the given Synset , is present in the sentence. The predicate $\text{has_Hyperonym}(\text{Hyperonym})$ succeeds if a word belongs to the Hyperonym . For example, given the sentence “*Let’s dance the war dance*” the predicate “ $\text{hasHyperonym}(\text{ASynset})$ ” would succeed for $\text{ASynset} = \text{synset of ritual dance}$ but not for $\text{ASynset} = \text{synset of social dancing}$. In order to improve accuracy and to decrease learning time further improvement has been performed based on the following observations:

- The word in instrumental case is usually a noun or an associated determinant. The noun to which the determinant is associated is usually the first noun to the left from this determinant. The determinant does not modify the classification.
- When the word in instrumental case is a noun (or determinant) it defines a relation between the noun and the nearest verb to the right.

Accuracy of finding the most significant words can be seen in Table 3. Then the semantic predicates are applied only to these important words.

Table 3: Finding the most significant words.

Type	Hit	Fail	Unknown	Total	Accuracy
Verb:	108	5	27	140	77.4
Noun:	105	7	0	112	93.8

There is a description of the new predicates.

- $\text{nearestNounNotVerbNotDet}/1$: looks for a word with at least one noun reading and which has no determinant neither verb readings. It first looks in the position, then in the left context and then in the right context, returning the first word found. For example, in the sentence *Etxe (house) batez (of a) jabetu (become the owner)* the goal $\text{nearestNounNotVerbNotDet}(\text{Word})$ success only for $\text{Word} = \text{house}$.
- $\text{nearestVerbNotDet}/1$: looks for a word with at least one verb reading and which has no determinant readings. It first looks in the position, then in the right context and then in the left context, returning the first word found. Using the same sentence as above as an example the predicate $\text{nearestVerbNotDet}(\text{Word})$ success only for $\text{Word} = \text{jabetu}$.

The following combinations of the semantic predicates and the morphological predicates were used:

- In all cases only synsets and the first level in the hyperonymy hierarchy are used, i.e. only $\text{nearestVerNotDetHasHyperonym}/1$, $\text{nearestNounNotVerbNotDetHasHyperonym}/1$, $\text{nearestVerbNotDetHasSynset}/1$ and $\text{nearestVeryNotDetHasHyperHyperonym}/1$ predicates are provided as background knowledge.
- In the first case $\text{exists}/1$ and $\text{forall}/1$ are used. The accuracy in this case is 56 %.

- In the second case only *exists/2* is used. This reduction is due to the available machine resources. In this case the accuracy is 59 %.

It demonstrates that the WordNet predicates do provide some valuable information, even when applied to ambiguously tagged text.

4.3 Refinement of the semantic predicates

When introducing the WordNet predicates a new source of ambiguity has been introduced. The words in the context are not semantically disambiguated and for that reason we can not remove any synset. Because of that we have to add all the possible semantic interpretations that a word can bear. In this section we explain how we tried to overcome this problem.

The problem cannot be completely solved without manual disambiguation. However, some improvements can be done if using the frequency of a given synset as measure for its “goodness”. We make the assumption that if two different words have a common synset (or hyperonym) it is more likely that this synset(hyperonym) is the right one. For example, given the sentences “sitting on the chair” and “sitting on the bank” we assume that the correct interpretation of “bank” is that of “chair” and not that of “credit institution”. We implement this in our application by removing all the synsets which appear less than N times in the database.

If to compare with the results displayed in the previous paragraph, in the first case the accuracy 57% while in the second it is about 60%. So there is slight improvement about 1 %. It is important that also the learning time was a bit smaller. It can be important when processing bigger data.

4.4 Leaving implicit class

The result of the last experiment is in Table 4. We can see that the biggest discrepancy between expected classification and the learned one concerns the class `instrument`. When we have a look to a typical result of learning (below) we can see that the default category (i.e. category that is used if no rule fires for the classified example) is again `instrument`. An example of output of Tilde is below.

```
class([time]):-nearestNounNotVerbNotDetHasHyperonym(s09065837),!.
% 6.0/6.0=1.0.
class([theme]):-nearestVerbNotDetHasSynset(s00527673),!.
% 13.0/13.0=1.0.
...
class([instrument]).
```

% 12.0/22=0.545

So we decided to remove the last clause from the learned rules. On one side it results in decrease of recall, in other side accuracy increased up to 10%. Namely for the two cases mentioned the accuracy increased up to 70.4% (recall 69.0) and 71.1% (recall 68.3).

Table 4: Results with arity 2 morphological predicates and refined WordNet.

REAL / PRED	cause	containing	instrument	manner	matter	theme	time	total
cause	0	0	1	1	0	3	0	5
containing	0	21	2	0	0	0	0	23
instrument	0	3	7	15	1	3	1	30
manner	0	1	5	29	1	5	0	41
matter	0	3	1	1	1	1	0	7
theme	1	1	1	4	0	23	0	30
time	0	0	0	0	0	1	5	6
total	1	29	17	50	3	36	6	142

5 Learning with Timbl

Timbl² (Zavřel & Daelmans, 1998) is a program implementing several instance-based, or Memory-Based, learning techniques. Timbl stores a representation of the training set explicitly in memory, and classifies new cases by extrapolation from the most similar stored cases.

5.1 Learning data

The propositional representation available for ILP had to be re-coded into the format required for Timbl. First, morphological information has been removed and the WordNet predicates have been rebuilt. In Timbl, each example is seen as a chain of comma separated items. So for each word in a given sentence, one of its lemmas is randomly chosen and the word/lemma pairs are written in a comma separated list of a given length. The *word in position* is always on the 5th position of the chain. As it may happen, that not all the examples are long enough, the missing ones are filled with underline characters. The category comes after the chain followed by a dot. An example is shown in Figure 3.

Figure 3: Two sentences in the Timbl format.

```
_,_,_,_,_,_,zauriz,zauri,betea,bete,_,_,_,_,_,containing.  
_,_,_,_,_,_,gauaz,gau,zaintzen,zaindu,zizkiena,edun,_,_,_,_,time.
```

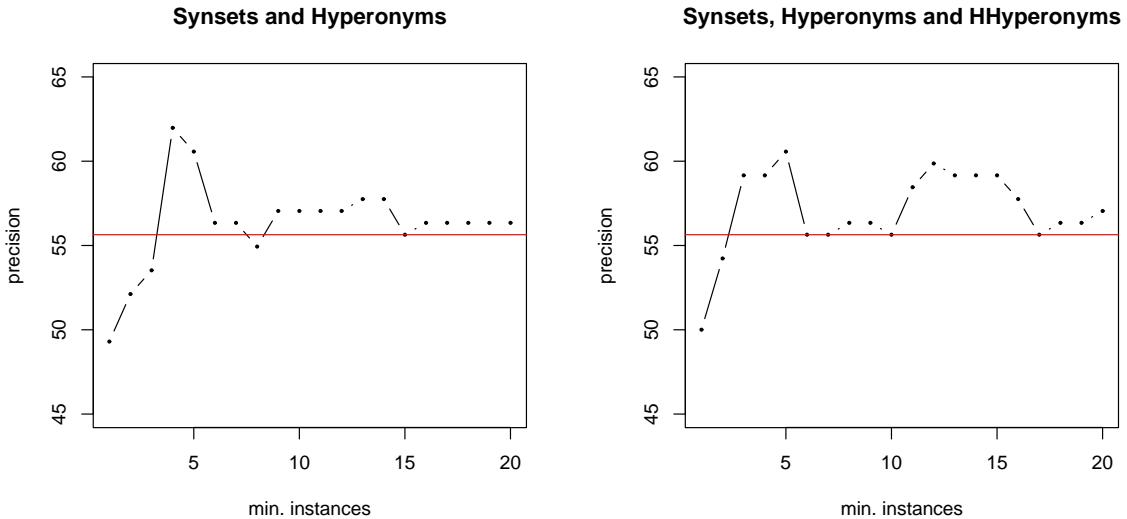
For each column pair (word/lemma) the union of the synsets of each line is found. If this set contains N elements, then N extra binary attributes are added to the database. For a given example the value of one of these binary attributes is true if and only if the synset it represents is a synset of the word in the given column.

5.2 Results

Two different set of experiments have been performed. They differ in the depth employed in the hyperonymy tree. In the first case only synsets and hyperonyms are considered. In the second

²<http://ilk.kub.nl/software.html>

Figure 4: Results for Timbl



one hyper-hyperonyms have also been used. Timbl was learned five times. In the first time the database is used without modification. In the next ones the synsets/hyperonyms which are true (following the schema above) for less than N examples are removed, for $N \in \{1..20\}$. The results can be seen in Figure 4. In both graphs, the horizontal line in the middle shows accuracy 55.6%, the case when no WordNet information has been exploited.

6 Related work

Agirre et al. (Agirre *et al.*, 2002) presented preliminary experiments in the use of translation equivalences to disambiguate prepositions or case suffixes. The core of the method is to find translations of the occurrence of the target preposition or case suffix, and assign the intersection of their set of interpretations. Given a table with prepositions and their possible interpretations, the method is fully automatic. The method was tested on the occurrences of the Basque instrumental case -z in the definitions of a Basque dictionary, looking for the translations in the definitions from 3 Spanish and 3 English dictionaries. The method is able to disambiguate with 94.5% accuracy 2.3% of those occurrences (up to 91). The ambiguity is reduced from 7 readings down to 3.1.

There has been many works that apply ILP for morphological disambiguation. Cussens (Cussens, 1997) developed POS tagger for English that achieved per-word accuracy of 96.4 %. Eineborg and Lindberg induced constraint grammar-like disambiguation rules for Swedish with the accuracy of 98%. In (Džeroski & Erjavec, 1997) ILP was applied for generating the lemma from the oblique form of nouns as well as for generating the correct oblique form from the lemma, with the average accuracy 91.5 %. Learning nominal inflections for Czech and Slovene (among others) is described in (Manandhar *et al.*, 1998). In (Cussens *et al.*, 1999), first steps in morphosyntactic tagging of Slovene are described. The obtained accuracy 86.6% is comparable with our results of tag disambiguation that varied between 80% and 98%. In (Nepil *et al.*, 2001; Žáčková & Popelínský, 2000) we brought first results for morphological tagging in Czech with

means of ILP. We did not employ any lexical statistics and we did not use any hand-crafted domain knowledge.

7 Conclusion

The results are not good enough for automatic disambiguation of cases in Basque. However, some conclusions can already be made. When using only simple morphological predicates an accuracy varied between 47% and 55%. When we introduce semantic predicates they produce a small improvement 59%. We can improve these results a little bit more by refining the semantic predicates trying to remove ambiguity as described in Section 4.3. When removing the implicit rule we reached accuracy higher than 70% with decrease of recall to 68–69%.

This fact seems to confirm that the information derived from the WordNet is important but does not mean that the morphological information should be automatically discarded. In the experiments described in Section 4.2 the morphological information is important for finding the so called "important" words (the words for which the WordNet predicates are applied). The morphological information is also used in *verbInPosition* and *adverbInPosition* predicates. Nevertheless we got the best results making no use of the morphological knowledge in the experiments described in Section 5.

Regarding to the experiments with Timbl, trying to find an explanation for the particular form the two curves show in Figure 4 is interesting. When we add the WordNet information the accuracy falls down about 6% in both cases and then it increases steadily to meet its peak value for $N \approx 5$. At this point the tendency changes and accuracy becomes worse. There is a possible explanation. When we added the synset information to the data we also add a lot of noise. By adding all the synsets of a word the only thing we do is adding all the possible semantic interpretations of a given word. When we restrict the minimum number of examples in which a synset must be present (the N parameter) data become less ambiguous. Those synsets which belong to different words have better chances to survive. The accuracy increases until we begin to destroy more information than noise. As N moves from 1 to 20 there is a balance between noise and information. The first peak could be due to the situation in which the synset and hyperonym information weigh more than the ambiguity they introduced. From that moment we begin to destroy information, so the curve sinks. The second peak belongs to the hyper-hyperonyms, which should be more common and thus are removed later. When this happens the second peak collapses.

Acknowledgements

Most of this work was done when Victor Lascurain and Luboš Popelínský worked with machine learning group at University of Freiburg, Germany. We would like to thank to Luc De Raedt and Stefan Kramer for their kind assistance. Luboš Popelínský has been partially supported by the grant of Czech Ministry of Education MŠMT 143300003. Eneko Agirre and Mikel Lersundi have been partially supported by the European Commission (MEANING project IST-2001-34460) and MCYT (Hermes project TIC-2000-0335-C03-03).

References

- AGIRRE E., LERSUNDI M. & MARTÍNEZ D. (2002). A multilingual approach to disambiguate prepositions and case suffixes. In *ACL Workshop: Word Sense Disambiguation: recent successes and future directions*.
- BLOCKEEL H. & RAEDT L. D. (1997). *Top-down Induction of Logical Decision Trees*. Rapport interne, Katholieke Universiteit Leuven. Department of Computer Science.
- CUSSENS J. (1997). Part-of-speech tagging using Progol. In *Inductive Logic Programming: Proceedings of the 7th International Workshop (ILP-97)*. LNAI 1297, p. 93–108: Springer.
- CUSSENS J., DŽEROSKI S. & ERJAVEC T. (1999). Morphosyntactic tagging of Slovene using Progol. In S. DŽEROSKI & P. FLACH, Eds., *Inductive Logic Programming: Proc. of the 9th International Workshop (ILP-99)*, Bled, Slovenia: Springer-Verlag.
- CUSSENS J. & (EDS.) S. D. (2000). *Learning language in Logic*. Springer.
- DŽEROSKI S. & ERJAVEC T. (1997). Induction of Slovene nominal paradigms. In *Inductive Logic Programming: Proceedings of the 7th International Workshop (ILP-97)*. LNAI 1297, p. 141–148: Springer.
- LAVRAČ N. & DŽEROSKI S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.
- MANANDHAR S., DŽEROSKI S. & ERJAVEC T. (1998). Learning multilingual morphology with CLOG. In *Inductive Logic Programming: Proceedings of the 8th International Conference (ILP-98)*: Springer.
- MUGGLETON S. (1992). *Inductive Logic Programming*. Academic Press.
- MUGGLETON S. & DE RAEDT L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, **19/20**, 629–679.
- NEPIL M., POPELIINSKY L. & ŽÁČKOVÁ E. (2001). Part-of-speech tagging by means of shallow parsing, ilp and active learning. In *Proceedings of the Third Learning Language in Logic (LLL) Workshop, Strasbourg, France*.
- NIENHUYSEN-CHENG S.-H. & DE WOLF R. (1997). *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.
- SARASOLA I. (1996). *Euskal Hiztegia*. Gipuzkoako Kutxa.
- ŽÁČKOVÁ E. & POPELÍNSKÝ L. (2000). Automatic tagging of compound verb groups in Czech corpora. In *Text, Speech and Dialogue: Proceedings of TSD'2000 Workshop, LNAI*: Springer.
- ZAVŘEL J. & DAELMANS W. (1998). *Recent Advances in Memory-Based Part-of-Speech Tagging*. Rapport interne, ILK/Computational Linguistics, Tilburg University.

Extracting XML syntactic chunks from Portuguese corpora

Caroline Gasperin(1), Renata Vieira(1), Rodrigo Goulart(1), Paulo
Quaresma(2)

(1) PIPCA - Unisinos
São Leopoldo, Brazil

{caroline,renata,rodrigo}@exatas.unisinos.br
(2) UEVORA
Evora, Portugal
pq@di.uevora.pt

Abstract

The Portuguese language has a great number of speakers distributed on Europe, South America, Asia and Africa but research and development on Portuguese processing are still limited compared to languages such as English, French and Spanish. Regarding parsing tools, one basic component for NLP systems, the CURUPIRA parser (Martins, 2002) and the PALAVRAS parser (Bick, 2000) have been recently made available. The PALAVRAS parser is a robust tool and we have used it as a basis in the development of previous work on Portuguese NLP applications. As a way to promote the applicability of this tool we propose an XML encoding for its output. According to our proposal, linguistic information may be provided in XML and it can be tailored to the needs of different NLP application. The paper illustrates the use of our tool and schemes by describing two applications that make use of different linguistic information extracted from Portuguese parsed corpus.

1 Introduction

The Portuguese language has a great number of speakers but the availability of computational tools is limited, specially, regarding parsing tools. (A list of existing tools for Portuguese can be found at <http://www.linguateca.pt/ferramentas.html>.) We have the CURUPIRA parser available since 2002 (Martins, 2002), and the PALAVRAS parser since 2000 (Bick, 2000). PALAVRAS is a parser based on constraint-grammar formalism developed at the Institute of Language and Communication of the University of Southern Denmark. It is available on the Internet¹ and is a robust tool. Still, one problem we find when using the analyses provided by PALAVRAS is that it is not in a standard format, so the extraction of syntactic information from parsed corpora depends on specific tools that have to be built for each intended application.

In this paper we present our tool and schemes for XML encoding the output of PALAVRAS. The parser has been used as a basis for previous work related to Portuguese processing (Vieira et al., 2000; Gasperin *et al.*, 2001). In these works, different tools for extracting syntactic

¹<http://visl.hum.sdu.dk/visl/pt>

```

STA:fcl
SUBJ:np
=>N:num('três' M P <card>) Três
=H:n('acidente' M P) acidentes
=N<:adj('grave' M P) graves
P:v-fin('marcar' PS/MQP 3P IND) marcaram
ACC:np
=>N:art('o' <artd> M S) o
=H:n('fim_de_semana' M S) fim_de_semana
.

```

Figure 1: Parsed sentence.

information had to be developed, adding time costs to the projects. In order to make the use of syntactic information from parsed corpora a simpler task, we developed a tool that generates the XML encoding for the PALAVRAS output. Our tool can also extract specific chunks from the parsed corpora according to the linguistic information needed for different NLP tasks. In this paper we illustrate the extraction of chunks for different applications: we use lists of NPs for anaphora resolution and triples of subject-verb-object to acquire knowledge from texts.

The paper is organized as follows. In Section 2 we present the parser output format. Section 3 presents our XML encoding principles. Section 4 presents how we generate XML output and extract chunks from the parsed corpus. Section 5 shows the use of our tool in different NLP applications. Our concluding remarks are presented on section 6.

2 PALAVRAS output

Take the sentence in Portuguese “Três acidentes graves marcaram o fim de semana.” (Three serious accidents marked the weekend.). The parser output for this sentence is shown on Figure 1.

On each line of the figure, the first symbol represents the syntactic function ('SUBJ'=subject, 'N'=noun modifier, 'H'=head, 'P'=predicator, 'ACC'=direct object); after the ':' there is the syntactic form for groups of words and POS-tags for single words ('np'=noun phrase, 'n'=noun, 'v'=verb, etc.); in brackets there is the word canonical form and other inflectional tags; after the brackets there is the word as it occurs in the corpus. The '=' signs in the beginning of each line represent the level of the phrase in the parsing tree.² Because this is not a standard format, the extraction of syntactic information from analysed corpora requires parsing it for different NLP applications. To simplify the use of parsed corpora, we transform PALAVRAS output into XML chunks. In this way we can use the XML tools already available to access the information that is needed.

3 Encoding principles

Our proposal is mainly influenced by MMAX (Müller & Strube, 2001b), the annotation tool that we have been using for several experiments on corpus annotation, as reported in (Vieira *et al.*, 2002b; Salmon-Alt & Vieira, 2002; Vieira *et al.*, 2002a). In (Müller & Strube, 2001a)

²A complete description of the tagset symbols is available at <http://visl.hum.sdu.dk/visl/pt/info/symbolset-manual.html>.

```

<!ELEMENT words (word*)>
<!ELEMENT word (#PCDATA)>
<!ATTLIST word
  id ID #REQUIRED
>
<!ELEMENT text (paragraph+)>
<!ELEMENT paragraph (sentence*)>
<!ATTLIST paragraph
  id ID #REQUIRED
>
<!ELEMENT sentence (EMPTY)>
<!ATTLIST sentence
  id ID #REQUIRED
  span CDATA #REQUIRED
>
(a)                                (b)

<!ELEMENT markables (markable*)>
<!ELEMENT markable (#PCDATA)>
<!ATTLIST markable
  id ID #REQUIRED
  span CDATA #REQUIRED
  type CDATA #REQUIRED
  member CDATA #IMPLIED
  pointer IDREF #IMPLIED
>
(c)

```

Figure 2: MMAX DTDs.

the following encoding architecture is presented. There is a base input file that describes the corpus tokens codified as `<word>` elements, according to the DTD presented in Figure 2(a). A second input file identifies the text structure (paragraphs and sentences), according to Figure 2(b). The output file contains the annotation done over the corpus. The annotation is codified by `<markable>` elements according to the DTD shown in Figure 2(c). Other attributes can be specified by the user according to his own annotation task.

Since we intend to follow standards for corpora annotation (Ide & Romary, 2002), we adopted the words file as proposed by (Müller & Strube, 2001a) as our basic file to which every other linguistic information should refer to. The words file for our example is shown on Figure 3.

In our scheme we identify syntactic structures as `<chunk>` elements into the chunks file (whose DTD is an extended version of the text structure DTD) and additional POS information is described in a POS file, both referring to the basic words file. Next section presents these files in detail.

4 Extracting chunks from parsed corpora

The program that transforms the output of the parser into XML, first generates Prolog terms corresponding to each parsed sentence. Figure 6 shows the terms for the parsed sentence in Figure 1.

From the Prolog terms the following files are generated: a basic words file, a POS file, and the chunks file which is tailored according to the application needs. We can indicate the kind of chunks to be extracted, just informing it as parameters of a Prolog predicate. For example, if we intend to extract noun phrases, we need to inform the program the value “np” as parameter.

The XML chunks are specified according to the DTD shown on Figure 4. The syntactic function of a chunk is given by its *function* attribute. The attribute *form* corresponds to the syntactic form of a word group or to the POS category of a single word. The chunks *span* attribute refers to `<word>` elements of the basic file. Figure 5 shows the complete chunks file for our example

```

<!ELEMENT text (paragraph+)>
<!ELEMENT paragraph (sentence*)>
<!ATTLIST paragraph
  id ID #REQUIRED
>
<!ELEMENT sentence (chunk*)>
<!ATTLIST sentence
  id ID #REQUIRED
  span CDATA #REQUIRED
>
<!ELEMENT chunk (chunk*)>
<!ATTLIST chunk
  id ID #REQUIRED
  function CDATA #REQUIRED
  form CDATA #REQUIRED
  span CDATA #REQUIRED
>

```

<words>

<word id="word_1">Três</word>

<word id="word_2">acidentes</word>

<word id="word_3">graves</word>

<word id="word_4">marcaram</word>

<word id="word_5">o</word>

<word id="word_6">fim_de_semana</word>

<word id="word_7">>.</word>

</words>

Figure 3: Words file.

Figure 4: Chunks DTD.

```

<text>
  <paragraph id="paragraph_1">
    <sentence id="sentence_1" span="word_1..word_14">
      <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
        <chunk id="chunk_2" function="n" form="num" span="word_1"/>
        <chunk id="chunk_3" function="h" form="n" span="word_2"/>
        <chunk id="chunk_4" function="n" form="adj" span="word_3"/>
      </chunk>
      <chunk id="chunk_5" function="p" form="v_fin" span="word_4"/>
      <chunk id="chunk_6" function="acc" form="np" span="word_5..word_6">
        <chunk id="chunk_7" function="n" form="art" span="word_5"/>
        <chunk id="chunk_8" function="h" form="n" span="word_6"/>
      </chunk>
    </sentence>
  </paragraph>
</text>

```

Figure 5: Complete chunks file.

```

sentence(syn(
  sta(fcl),
  subj(np,
    n(num('três','M','P','<card>'),'Três'),
    h(n('acidente','M','P'),'acidentes'),
    n(adj('grave','M','P'),'graves')),
    p(v_fin('marcar','PS/MQP','3P','IND'),'marcaram'),
    acc(np,
      n(art('o','M','S'),'o'),
      h(n('fim_de_semana','M','S'),'fim_de_semana', '.')))).
```

Figure 6: Prolog terms.

```

<!ELEMENT words (word*)>
<!ELEMENT word (n|prop|adj|v|art|pron|adv|num|
  prp|intj|conj)>
<!ATTLIST word id ID #REQUIRED>

<!ELEMENT n (secondary_n?)>
<!ATTLIST n
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT prop (secondary_prop?)>
<!ATTLIST prop
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT adj (secondary_adj?)>
<!ATTLIST adj
  canon CDATA #REQUIRED
  gender (M | F) #REQUIRED
  number (P | S) #REQUIRED
>

<!ELEMENT v ((fin|inf|pcp|ger), sec-
ondary_v?)>
<!ATTLIST v canon CDATA #REQUIRED>
<!ELEMENT fin EMPTY>
<!ATTLIST fin
  person (1S|2S|3S|1P|2P|3P) #RE-
QUIRED
  tense (PR|IMPF|PS|FUT|IMP) #RE-
QUIRED
  mode (IND|SUBJ) #REQUIRED
>
<!ELEMENT inf EMPTY>
<!ELEMENT pcp EMPTY>
<!ATTLIST pcp
  gender (M|F) #REQUIRED
  number (P|S) #REQUIRED
>
<!ELEMENT ger EMPTY>
...

```

```

<words>
  <word id="word_1">
    <num canon="três" gen-
    der="M" number="P">
      <secondary_num tag="card"/>
    </num>
  </word>
  <word id="word_2">
    <n canon="acidente" gen-
    der="M" number="P"/>
  </word>
  <word id="word_3">
    <adj canon="grave" gen-
    der="M" number="P"/>
  </word>
  <word id="word_4">
    <v canon="marcar">
      <fin tense="PS/MQP" per-
      son="3P" mode="IND"/>
    </v>
  </word>
  <word id="word_5">
    <art canon="o" gender="M" num-
    ber="S">
      <secondary_art tag="artd"/>
    </art>
  </word>
  <word id="word_6">
    <n canon="fim_de_semana" gen-
    der="M" number="S"/>
  </word>
</words>

```

Figure 8: Words POS file.

Figure 7: Words POS DTD.

sentence.

The POS file is specified according to the DTD shown on Figure 7. This DTD was based on the PALAVRAS tag set. For our example sentence, we have the POS file shown on Figure 8.

5 Using XML chunks

In this section we illustrate briefly how we use Portuguese syntactic chunks in two different applications: anaphora resolution and knowledge extraction from texts. Both applications are at early stages of development, our intention in this section is mainly to show how the chunks can serve different purposes, rather than discuss results related to these applications. We believe that the examples may help other users interested in using the tool for other applications.

```

<paragraph "paragraph_1">
  <sentence id="sentence_1" span="word_1..word_14">
    <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
      <chunk id="chunk_2" function="h" form="n" span="word_2"/>
    </chunk>
    ...
  </sentence>
</paragraph>

```

Figure 9: NP chunks

5.1 Anaphora and coreference resolution

We are developing a multi-lingual tool for anaphora resolution of definite descriptions (...*the boys*...), demonstrative noun phrases (...*these boys*...) and pronouns (...*they*...). The main goal is to identify the antecedents for these anaphoric expressions. As we are considering those cases where the antecedent is a noun phrase, we need to extract all NPs from the parsed corpus. For extracting NP chunks, we select chunks whose syntactic form is “np”. Figure 9 shows NP chunks for our example sentence.

From NP chunks, we select anaphors and antecedent candidates. Our anaphors are identified by the presence of definite article, demonstrative and personal pronouns. This information is given in the POS file. Then, we apply heuristics to identify the correct antecedent among the candidates. The heuristics to be used are based on previous studies about resolution of nominal referring expressions (Vieira & Poesio, 2000; Lappin & Leass, 1994; Strube *et al.*, 2002) and they are not discussed here. These tasks are performed by a set of stylesheets. Each one is connected to another through pipes and it filters the information flowing through the system (Gamma *et al.*, 1995). There are three main steps: anaphor selection, candidates selection and markables generation (Figure 10 A, B and C respectively). Each step corresponds to one stylesheet.

The Anaphor selection task (A) receives chunks and uses POS information to select the anaphors. In the selection a node `<anaphor>` is created, it contains the *span* attribute referring to the words file, and another node `<header>` contains the head noun of the NP (Figure 11(a)).

The Candidates selection task (B) selects antecedent candidate from the NP chunks (as shown in Figure 11(b)).

Finally, the Markables generation task (C) matches head nouns using the `<anaphor>` and `<candidate>` information through the application of a sequence of heuristics informed in the rule base. The output is MMAX compatible, so the results can be visualized using the MMAX tool. Figure 12 shows the output markables.

5.2 Knowledge acquisition from texts

Our second application is related to experiments towards semi-automatic generation of conceptual maps from a parsed corpus. From parsed texts, we first extract triples of subject-verb-object. Over these triples we apply a set of filtering heuristics based on the frequency of the relations and frequency of the terms. We are also investigating whether different terms appearing in the same relations of other terms form a set of semantically related words. For extracting subject,

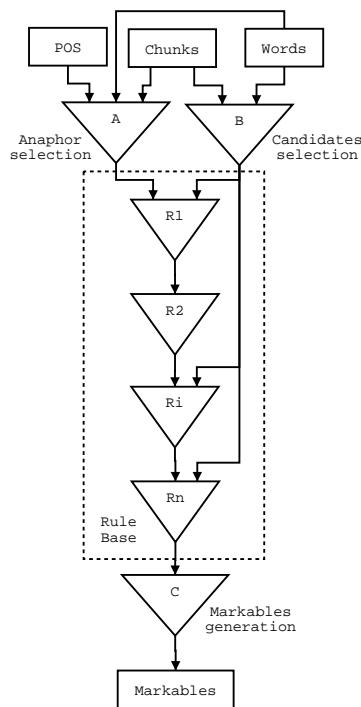


Figure 10: Anaphora resolution design

```

<anaphor span="word_5..word_8">
  <header>final</header>
</anaphor>

```

(a)

```

<candidate span="word_1..word_3">
  <header>acidentes</header>
</candidate>
<candidate span="word_5..word_6">
  <header>final</header>
</candidate>

```

(b)

Figure 11: Anaphor nodes (a) and Candidate nodes (b)

```

<markables>
  <markable id="markable_1" pointer="" span="word_5..word_8" classification="discourse_new"/>
</markables>

```

Figure 12: Markable nodes

```

<paragraph id="paragraph_1">
  <sentence id="sentence_1" span="word_1..word_9">
    <chunk id="chunk_1" function="subj" form="np" span="word_1..word_3">
      <chunk id="chunk_2" function="h" form="n" span="word_2"/>
    </chunk>
    <chunk id="chunk_3" function="p" form="v_fin" span="word_4"/>
    <chunk id="chunk_4" function="acc" form="np" span="word_5..word_8">
      <chunk id="chunk_5" function="h" form="n" span="word_6"/>
    </chunk>
  </sentence>
</paragraph>

```

Figure 13: Subject and Object chunks

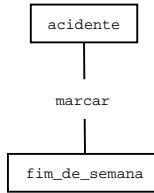


Figure 14: Example of a relation in a conceptual map.

verb and object chunks, we select that ones with *function* equal to “subj”, “acc”, and *form* equal “vp”, “v” and “np”.

For our example case, we consider (Figure 13) chunk_2 (subject head noun), chunk_3 (verb) and chunk_5 (object head noun), generating the triple “acidente-marcar-fim_de_semana”. Triples are extracted from chunks of the whole corpus; after filtering the set of triples relations between two concepts, a conceptual map is generated, as shown in Figure 14. We use the CMap tool (<http://cmap.coginst.uwf.edu/>) to generate the map. The heuristics for filtering the triples are being defined.

This methodology has been applied to a subset of the Portuguese Attorney General’s Office documents (Quaresma & Rodrigues, 2003). We have selected a subset of 40 documents having event descriptions. These documents were parsed and the correspondent XML chunks (subject, verb, object) were produced. The triples and the correspondent conceptual maps were created and are currently under analysis. These conceptual relations may be also used for the creation of an ontology of actions. In previous work (Saias & Quaresma, 2002) it is shown how to automatically transform conceptual relations into an ontology defined using the DAML+OIL/OWL semantic web language (DAM, 2000).

6 Concluding remarks

In this paper we presented a tool that extracts XML chunks from the PALAVRAS parser output. The chunks generated by our tool may reflect both the complete parsing and selected information tailored to a particular application. With the XML encoding of the linguistic information, different NLP applications can access it through already available XML tools.

Besides the advantages of using XML, our schemes are compatible with an existing annotation tool, MMAX. We have also presented two applications developed on the basis of the tool and

schemes presented here.

As current work we are adapting our encoding schemes to proposed standards (Ide & Romary, 2002). We are also developing a web interface to integrate our tools to the PALAVRAS parser. With this work we intend to promote the access and use of basic tools for the development of NLP applications for the Portuguese language.

Acknowledgments

We would like to thank CNPq(Brazil)/INRIA(France) and CAPES(Brazil)/FCT(Portugal) for their financial support. We are grateful to Eckhard Bick for his valuable help on the use of the parser PALAVRAS, Christoph Müller and Michael Strube for providing background for our annotation schemes, and Susanne Salmon-Alt.

References

- (2000). *DAML+OIL – DARPA Agent Markup Language*. DAML, <http://www.daml.org>.
- BICK E. (2000). *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Århus University, Århus.
- GAMMA E., HELM R., JOHNSON R. & VLISIDES J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. New York: Addison-Wesley Publishing Company.
- GASPERIN C., GAMALLO P., AGUSTINI A., LOPES G. & LIMA V. (2001). Using syntactic contexts for measuring word similarity. In *Proceedings of the Workshop on Semantic Knowledge Acquisition and Categorisation*, Helsinki, Finland.
- IDE N. & ROMARY L. (2002). Standards for language resources. In *Proceedings of the LREC 2002*, p. 839–844, Las Palmas de Gran Canaria.
- LAPPIN S. & LEASS H. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, **20**(4).
- MARTINS, R. T.; HASEGAWA R. N. M. (2002). *Curupira: um parser funcional para o português*. Nilc-tr-02-06, USP-UNESP-UFSACAR, São Carlos.
- MÜLLER C. & STRUBE M. (2001a). Annotating anaphoric and bridging expressions with MMAX. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*, p. 90–95, Aalborg, Denmark.
- MÜLLER C. & STRUBE M. (2001b). MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the IJCAI 2001*, p. 45–50, Seattle.
- QUARESMA P. & RODRIGUES I. P. (2003). PGR: Portuguese attorney general's office decisions on the web. In BARTENSTEIN, GESKE, HANNEBAUER & YOSHIE, Eds., *Web-Knowledge Management and Decision Support*, Lecture Notes in Artificial Intelligence LNCS/LNAI: Springer-Verlag. To be published.

- SAIAS J. & QUARESMA P. (2002). Semantic enrichment of a web legal information retrieval system. In T. BENCH-CAPON, Ed., *JURIX'2002 - Fifteenth Annual International Conference on Legal Knowledge and Information Systems*, London, UK: IOS Press.
- SALMON-ALT S. & VIEIRA R. (2002). Nominal expressions in multilingual corpora: Definites and demonstratives. In *Proceedings of the LREC 2002*, Las Palmas de Gran Canaria.
- STRUBE M., RAPP S. & MÜLLER C. (2002). The influence of minimum edit distance on reference resolution. In *Proceedings of the EMNLP 2002*, Philadelphia.
- VIEIRA R. & POESIO M. (2000). An empirically-based system for processing definite descriptions. *Computational Linguistics*, **26**(4), 525–579.
- VIEIRA R., SALMON-ALT S., GASPERIN C., SCHANG E. & OTHERO G. (2002a). Coreference and anaphoric relations of demonstrative noun phrases in multilingual corpus. In *Proceedings of the DAARC 2002*, Estoril.
- VIEIRA R., SALMON-ALT S. & SCHANG E. (2002b). Multilingual corpora annotation for processing definite descriptions. In *Proceedings of the PortAL 2002*, Faro.
- VIEIRA ET AL. (2000). Extração de sintagmas nominais para o processamento de co-referência. In *Anais do V Encontro para o processamento computacional da Língua Portuguesa escrita e falada - PROPOR*, Atibaia.

Example-based NLP for Minority Languages: Tasks, Resources and Tools

Oliver Streiter and Ernesto William De Luca
European Academy, 39100 Bolzano/Bozen, Italy
[{ostreiter;edeluca}](mailto:{ostreiter;edeluca}@eurac.edu)@eurac.edu

Mots-clefs – Keywords

langues minoritaires, traitement automatique des langues naturelles basé sur des exemples
minority languages, example-based NLP

Résumé - Abstract

Dans cet exposé nous analysons la relation entre le traitement automatique des langues minoritaires et les approches au Traitement des Langues Naturelles. Nous donnons un aperçu des tâches qui ont été affrontées et des approches utilisées. Vu que les ressources linguistiques sont limitées (telles que les dictionnaires et les corpus), le MLP emploie souvent des approches basées sur des règles, bien qu'elles demandent un investissement temporellement immense. L'approche statistique peut être plus efficace à condition que des corpora appropriés soient accessibles. Comme deuxième alternative nous présentons l'approche basée sur des exemples. L'avantage de cette approche est de nécessiter des ressources linguistiques plus petites et d'intégrer un module d'apprentissage. Nous démontrons que presque toutes les tâches usuelles du TALN peuvent être affrontées par cette approche. Des ressources linguistiques et des outils sont souvent librement disponibles.

In this paper we focus on the relation between **minority language processing** (MLP) and approaches to **Natural Language Processing** (NLP). We list the tasks which have been tackled in MLP and look at the approaches taken. Because there are few linguistic resources (such as dictionaries and corpora) MLP often makes use of **rule-based** (RB) approaches. However, RB-approaches are time-consuming to implement. The alternative statistical (S) approach has shown to be more efficient if appropriate corpora are available. We present the **example-based** (EB) approach as second alternative to RB-approaches, having the advantage of requiring smaller linguistic resources than statistical approaches and integrating a learning component. We show that actually most NLP tasks can be handled by EB-approaches. Linguistic resources and tools are often freely available.

1 Introduction

The automatic processing of minority languages (MLP) may require approaches and techniques different from those which are commonly used in NLP. Most research and development in NLP is done on a dozen of European and Asian languages which have incomparably more resources, in terms of human resources and software than small languages have (Somers, 1998). Minority languages tend to have few speakers, even fewer native linguists and very few computational linguists. The financial support for MLP-projects is often scarce or absent. Even the most basic computational infrastructure for minority languages may be absent: While important languages dispose of (free) corpora, tree-banks, parsers, taggers, morphological analyzers, 75% of all languages even have no standardized writing system (e.g. Mín Nán¹ or Haitian Creole²). At least in Europe, the situation is improving due to the support the EU grants to MLP-related projects.

In this paper we try to analyze the relation between NLP approaches and MLP. In Section 2.1 we give an overview of tasks for NLP and MLP. In Section 2.2 we introduce different approaches to handle these NLP tasks. The advantages of example-based approaches for MLP, hinted at in (Somers, 1997) for the task of Machine Translation, are explained: EB-approaches have the advantage to require smaller or diversified linguistic resources than statistical approaches do. These approaches integrate a learning component and thus allow for easy updates and improvements. EB-approaches can be applied to a great number of NLP tasks relevant for minority languages (Tab. 2). Steep learning curves, the possibility to start from very few examples and the possibility to handle exceptions are additional advantages.

In Section 2.3 we summarize the research done in MLP, focusing on the approaches taken. After a short discussion of why we might find this distribution of NLP approaches, we investigate the feasibility of the example-based (EB) approach for MLP by checking the availability of tools and ressources.

2 Research in NLP and MLP

2.1 Tasks, Resources and Tools

Speakers of minority language want to use their language in the same way as speakers of major languages do. They want to create documents, mail, chat, send short messages, use a word editor with facilities as spell and syntax checkers. Others may want to have news or scientific articles to be automatically translated into their languages. All these are typical *NLP tasks*. They may involve one or more sub-tasks. Syntax checking requires, for example, morphological analysis, POS-tagging, word sense disambiguation, phrase chunking and parsing. Table 1 lists NLP tasks which are either main tasks, invisible sub-tasks or tasks which are required in order to create an electronic linguistic resource.

NLP requires two types of resources, linguistic data in an electronic format and computational engines. Typical *electronic linguistic resources* are electronic dictionaries and corpora. Computational engines process the linguistic resources (the training material) and produce an output. These *tools* are mostly invisible to a user of an NLP application.

Table 1: NLP tasks and their possible functions.

NLP task	Function
concordancing	takes a document and creates a list of the sentences containing the query term in alphabetical or frequency order. Useful tool for linguists and language learners.
corpus construction	collects, formates, classifies and annotates documents for linguistic analysis and the training of NLP tools.
document classification	serves to provide information, such as the language, format, encoding, subject field, etc about a document. A simple classifier is the UNIX-command "file".
hyphenation	suggests word-internal positions for hyphenation, necessary for all word editors.
machine translation	automatically translates between natural languages.
morphological analysis	gives all possible analyses of an inflected, derived or compounded word.
OCR, Optical Character Recognition	transforms printed documents into electronic documents, important for corpus construction.
parsing	identifies the sentence structure. Used for style checking, machine translation, etc.
phrase chunking	identifies non-recursive phrases. Used after POS-tagging and before parsing.
POS-tagging	identifies the POS of a word in case the word is ambiguous.
sentence boundary detection	is needed for all NLP-tasks which have as input a text but work on sentences.
sign recognition	transforms deaf sign language gestures into a different representation.
spell checking	is an important step in the creation of a document; this step has an impact on the orthographic standardization of a language.
spelling conversion	is important for the creation of corpora in a specific spelling variant.
stemmer	strips suffix from a word. Used for information retrieval and spell checking.
term base	stores technical terms, explains the terms and illustrates their usage.
term extraction	helps in the creation of terminological data for specific sub-languages.
word sense disambiguation	helps with spell and syntax checking, document retrieval, machine translation.

2.2 Approaches to NLP

Approaches to NLP are traditionally classified into rule-base (RB), statistical (S) and example-based (EB) approaches. We reformulate this classification in view of the the greatest problem of MLP, i.e. the absence of linguistic resources.

Example-based (EB) approach The training material, i.e. the linguistic resource, of EB-approaches has the same internal complexity as the output. E.g. the training material consists of translations in order to create translations, of terms in order to extract terms, of correct spellings in order to correct the spelling etc.

The EB-approach tries to solve problems (to analyze a word or a sentence) by finding the solution (the analysis) of a problem most similar to the problem to be solved (a similar word or sentence). The old solution is left unmodified or is adapted and combined with other retrieved solutions in order to solve the new problem.

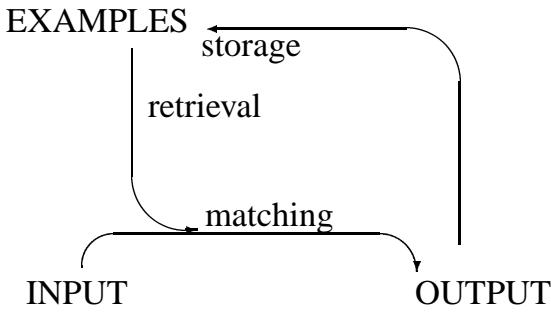
The EB-approach integrates a learning component. Input and output are of the same structure. Therefore the output may be added to the set of training examples and improve the accuracy of the system (Somers, 2001; Streiter, 2002b). A simplified illustration of the EB approach is the the spelling checker *ispell*.¹⁸ The system consists of examples of correctly written words. If a word is not found, the most similar words are searched and suggested as solutions. The system can be easily trained by adding unknown but correctly written words to the list of examples. As

Table 2: Example-based approaches to NLP Tasks

NLP task	Language
document classification	English ³ , German-Italian-Lad ⁴
hyphenation	Dutch ⁵
machine translation	English-French ⁶ , German-English ⁷ , Japanese-English ⁸
morphological analysis	English ⁹
parsing	Chinese ¹⁰ , German ¹¹
phrase chunking	German ¹²
POS-tagging	Dutch, Spanish, Swedish, German ¹³
sentence boundary detection	English ¹⁴
term extraction	Lad (Streiter <i>et al.</i> , 2002)
word sense disambiguation	Chinese ¹⁵
spell checking	(Afr, Cat, Gal, Iri, Wal) ¹⁶ , (Bre, Cat, Wel, Far) ¹⁷

Table 2 shows, most NLP tasks can be handled by EB-approaches.

Figure 1: The learning component as integral part of the EB-approach.

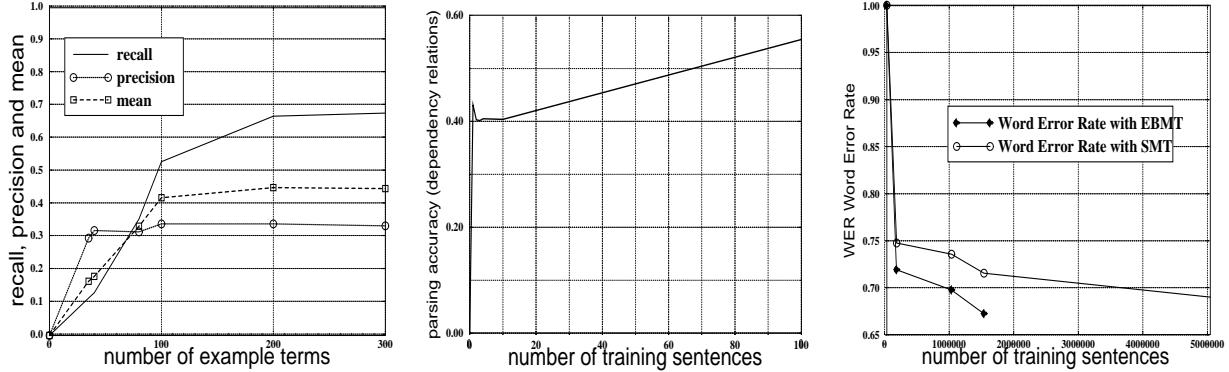


Even if a few examples are available, EB-approaches may obtain remarkably good results. The following data illustrated this fact: In a term extraction task for Ladin, with only 300 example terms (from a general purpose dictionary (Streiter *et al.*, 2002)) we obtain almost 70% recall on a Ladin document of about 1000 words. In a parsing experiment we parsed 300 untrained sentence in an EB-framework. One example tree is sufficient to get 43% of the dependency relations correct (Streiter, 2002a). In addition, the learning component in EB-approaches allows for the early investment of manually corrected examples (Day *et al.*, 1997). A statistically significant speed-up of this early investment could been proved, however, for very specific conditions only (Streiter, 2001). The relation between the quality of EB-machine translation and statistical machine translation and the size of the training corpus is analyzed in (Carl & Langlais, 2003): Due to the usage of richer data in EBMT, less data are required.

Statistical (S) approach Approach using internally less structured resources than the expected output. In order to compensate for this lack of information, larger quantities of linguistic resources are used. The internal complex structure is induced based on observed frequencies.

Rule-based (RB) approach Approach using resources which are more complex than the output. The output is deduced from the linguistic resource (rules). These rules are induced by linguists, usually on the basis of much smaller empirical data. Rules may comprise such complex structures as grammars, dictionaries, ontologies etc.

Figure 2: Example-based term extraction with very few example terms (left). Example-based parsing with very few example trees (mid). Word Error Rate with Example-based Machine Translation and Statistical Machine Translation (right).



Comparison of approaches In the absence of linguistic resources, RB-approaches seem to be most promising and the first intuitive choice. However, due to the only partial regularity of natural languages, highly trained linguists have to work for a long time in order to produce rule-sets which can handle NLP tasks. While learning-curves are initially steep with RB-approaches, they do not rise monotonically.

The statistical approach is the most popular alternative to the RB-approach. It requires large training corpora. The EB-approach is a second alternative and has been tested for various NLP tasks (Table 2). EB-based approaches may start with only 1 or 2 training items, according to the NLP task. EB-approaches require more structured linguistic resources than statistical approaches do (e.g. a treebank in place of a tagged corpus). Corpora for EB-approaches thus may be specific and limited in their further usage. Corpora used for statistical approaches are usually multi-purpose corpora. EB-systems may be developed half-automatically by processing a second, third, ... entity and adding the output to the set of examples. While RB-approaches face the difficulty to express the linguistic complexity completely and consistently, EB-approaches have no difficulties in handling exceptions (Daelemans *et al.*, 1999) because regular and irregular phenomena may be listed side by side in the example set. EB-approaches have, very much like RB-approaches, an initially steep learning-curve which not necessarily rises monotonically.

As a tendency, EB-systems are more complex to program in case the examples have to be combined and adapted. Statistical approaches require less programming efforts, but statistical language modeling instead. For both activities supporting tools are available.

2.3 Approaches to MLP

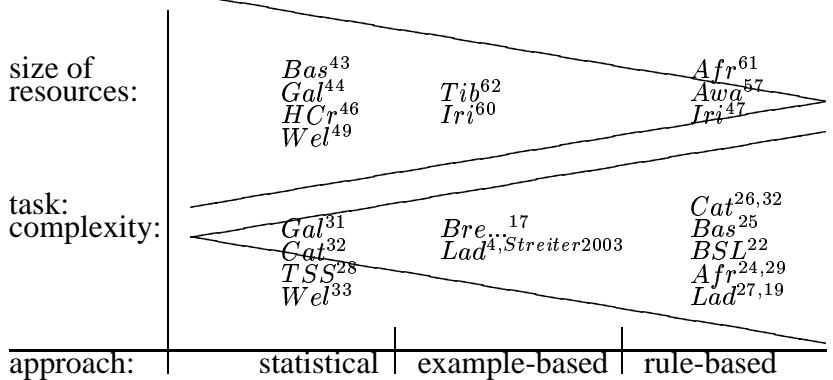
Table 3 links NLP-tasks to the approaches chosen in MLP. With the exception of a few languages, research is very punctual and far from offering solutions to more than a few NLP tasks. The choice of the approach may have been influenced by the task, the language, the time in which the research was done and personal preferences of researchers: While some tasks are preferred playground for specific approaches (statistical tagging, EB-spell checking and RB machine translation), some researchers, groups or languages have their specific preferences. The distribution of data, however, might also be interpreted as follows: Complex tasks are tackled with RB-approaches if linguistic resources are missing. Simple tasks may be handled with

Table 3: NLP tasks and corresponding approaches to Minority Language Processing

NLP task	Approach	Language
concordancer		Lad (Val di Fassa), Sar ¹⁹
corpus construction		Afr ²⁰ , Bas ²¹ , Lad , Sar ¹⁹
dictionary construction		Afr ²⁰
document classification	EB	Lad ⁴
hyphenation	EB	Cat, Gal (Latex)
machine translation	RB	Bas ²⁵ , BSL ²²
machine translation	RB+S	Min ²³
morphological analyzer	RB	Afr ²⁴ , Bas ²⁵ , Cat ^{26,32} , Lad (Fassa) ²⁷
OCR	unspecified	HCr ²
sign recognition	S	TSS ²⁸
spell checking	RB	Afr ²⁹ , Lad (Fassa) ²⁷
	EB ispell	(Afr, Cat, Gal, Iri, Wal) ¹⁶ , (Bre, Cat, Wel, Far) ¹⁷ , Fri ³⁰
tagging	S	Gal ³¹ , Cat ³²
spelling conversion	unspecified	HCr ²
stemming	RB	Afr ²⁹
term extraction	EB S	Lad (Ghardena) (Streiter <i>et al.</i> , 2002) Wel ³³
thesaurus, ontology	mixed	Iri (Scannell 2003) Bas ³⁴
term base		Lad (Val Badia, Ghardena) ³⁵ , Cat ³⁶ , Bas ³⁷

statistical approaches if corpora are available. Tasks of intermediate complexity are handled by EB-approaches, when corpus size does not compensate for the required complexity (c.f. Fig. 3).

Figure 3: Approaches to MLP and their dependence on resources and the complexity of the task.



3 Free Resources for Example-based NLP

In order to develop EB-NLP applications for minority languages, adequate tools and adequate linguistic resources are required. Table 4 lists a small set of freely available tools, ranging from general-purpose tools (TiMBL) to specific tools (e.g. MBT). If not marked otherwise, these tools can be downloaded from <http://ilk.kub.nl/software.html>.

Table 4: NLP tasks and freely available EB-NLP tools

NLP task	EB-NLP tools
general purpose	TiMBL (Tilburg Memory-Based Learner) ³⁸
POS-tagging	MBT (Memory-Based Tagger) ¹³
spell checking	myspell, ispell, aspell ³⁹
memory-based learner	Van den Bosch
document classifier	textcat ⁴⁰
term extractor	bistro ⁴¹
parser	OCTOPUS ⁴²

Table 5 finally lists some of the linguistic resources which are freely available for minority languages. Although these tables are not exhaustive, they show that the availability of resources is not comparable to what is available for the "major" languages. Nevertheless, interesting options become available: Dictionaries offer a rich set of examples for NLP tasks such as spell checking (Scannell, 2003), term extraction (Streiter *et al.*, 2002), named entity recognition and phrase chunking. For these tasks, the boundaries of words and phrases implicitly given in the dictionaries may be a sufficient "annotation". Annotated monolingual corpora provide examples, for hyphenation, sentence boundary detection, named entity recognition, morphological analysis, tagging, phrase chunking, word sense disambiguation and parsing. Parallel corpora can provide examples for translation-related NLP tasks.

Table 5: Free electronic linguistic resources for Minority Languages

Resource	Language
monolingual corpus	Bas(4.600.000 words) ⁴³ , Gal ⁴⁴ , Gal(180.000.000 words) ⁴⁵ , HCr(18.000.000 tokens) ⁴⁶ , Iri(12.000 words) ⁴⁷ , Wel(4.000.000 words) ⁴⁹ , Sar ⁵⁰
bilingual corpus	Gal-Portuguese ⁵¹ , Universal Declaration of Human Rights more than 300 Languages (small) ⁵²
sound corpus	Bas ⁵³ , Far(small) ⁵⁴
monolingual dictionary	Iri(small) ⁵⁵ Wel(small) ⁵⁵ , Lad (Val di Fassa) ⁵⁶
bilingual dictionary	Awa-English(small) ⁵⁷ , Bre-English-French ⁵⁸ , Bre-English(1451 entries) ⁵⁸ , HCr-English-French ⁵⁹ , Iri-English(14.000 words) ⁶⁰ , Afr-French(3000 entries) ⁶¹ , Tib-English (54.000) ⁶²
glossary	Gal ⁶³ , Gal-English(950 words) ⁶⁴
thesaurus	Iri (Scannell 2003)

4 Summary and Conclusions

In this paper we analyzed the relation between minority language processing and three different approaches to Natural Language Processing, the RB-approach, the statistical approach and the EB-approach. We investigated the possibility of using the EB-approach instead of RB- or statistical approaches, assuming that it might be easier and faster to list relevant examples than to write a coherent set of abstract rules on the one hand or to create large general-purpose corpora for statistical approaches on the other hand. The manual creation of examples can be avoided completely if examples are contained in linguistic resources such as dictionaries or corpora.

Tools for EB-approaches are freely available and well documented. EB-approaches have been

successfully applied to a great variety of NLP-tasks relevant for ML. All this seems to confirm our conviction that EB-approaches offer a processing strategy for MLs which merits to be examined in more detail.

Notes

- ¹<http://daiwanway.dynip.com/tw/writing.shtml>
- ² http://www.lisa.org/archive_domain/newsletters/2002/1.3/mason.html
- ³HAN & KARYPIS (2000). Centroid-based document classification: Analysis & experimental results.
- ⁴STREITER & VOLTMER (2002). Document classification for corpus-based legal terminology.
- ⁵DAELEMANS & VAN DEN BOSCH (1992). Generalisation performance of backpropagation learning on a syllabification task. In *TWLT3: Connectionism and Natural Language Processing*, p. 27–37, Enschede.
- ⁶BROWN (1996). Example-Based Machine Translation in the Pangloss System. In *COLING'96*.
- ⁷CARL & HANSEN (1999). Linking translation memories with example-based machine translation. In *MT-Summit'99*, Singapore.
- ⁸SATO & NAGAO (1990). Towards memory based translation. In *COLING'90*.
- ⁹VAN DEN BOSCH, DAELEMANS & WEIJTERS (1996). Morphological Analysis as Classification: an Inductive-Learning Approach. In *NeMLaP*, p. 79–89, Ankara.
- ¹⁰STREITER & HSUEH (2000) A case-study on example-based parsing. In *ICCLC2000*, Chicago.
- ¹¹KÜBLER (2003). *Memory-based Parsing of a German Corpus*. PhD thesis, University of Tübingen.
- ¹²TJONG KIM SANG (2001). Transforming a chunker to a parser. In *Comp. Linguistics in the Netherlands 2000*, Tilburg.
- ¹³DAELEMANS, ZAVREL, BERCK & GILLIS (1996). MBT: A memory-based part of speech tagger-generator. In *Fourth Workshop on Very Large Corpora*: University of Copenhagen.
- ¹⁴STEVENSON & GAIZAUSKAS (2000). Experiments on sentence boundary detection. In *1st Meeting of the North American Chapter of the ACL*, p. 24–30, Seattle.
- ¹⁵TONG, HUANG & GUO (1999). Example-based sense tagging of running Chinese text. In *Natural Language Processing Using Very Large Corpora*, Text, Speech and Language Technology.
- ¹⁶<http://fmg-www.cs.ucla.edu/geoff/spell-dictionaries.html>
- ¹⁷<http://aspell.sourceforge.net>
- ¹⁸In fact, *ispell* contains also a RB-component.
- ¹⁹GIULIANO (2002). A tool-box for lexicographers. In *EURALEX 2002*, Copenhagen.
- ²⁰<http://www.puk.ac.za/navorsing/eng/languages.html>
- ²¹SARASOLA (2000). Strategic priorities for the development of language technology in minority languages. In *LREC'2000 Workshop on Developing language resources for minority languages*, Athens.
- ²²MARSHALL & SÁFÁR (2002). Sign language generation using HPSG. In *TMI-2002*, Keihanna.
- ²³LIN & CHEN (1999). A Mandarin to Taiwanese Min Nan Machine Translation System with Speech Synthesis of Taiwanese Min Nan. in *International Journal of Computational Linguistics and Chinese Language Processing*.
- ²⁴STADLER & COETZER (1990). A Morphological Parser for Afrikaans. In *COLING'90*
- ²⁵DE ILARRAZA, MAYOR & SARASOLA (2000). Reutilización de recursos lingüísticos en la construcción de un sistema de ta inglés-euskara. In *IAI Working Paper No.36. Hybrid Approaches to Machine Translation*.
- ²⁶TONI BADIA ET AL. (1999). Catmorph, un analitzador morfològic par al tractament automàtic de corpus textuais en català. In *Llengua & Literatura*, 10, pp.329-360
- ²⁷BORTOLOTTI & RASOM (2003). The project Tales. In (Streiter *et al.*, 2003)
- ²⁸LIANG. A Real-time Continuous Gesture Recognition System for Taiwanese Sign Language. Thesis, NTU.
- ²⁹HUYSTEEN & VAN ZAANEN (2003). A Spellchecker for Afrikaans, based on Morphological Analysis. In *6th International Terminology in Advanced Management Application Conference (TAMA)*. Pretoria.
- ³⁰<http://www.fa.knaw.nl>
- ³¹VILARES FERRO ET AL. (1998). A tagger environment for Galician. In *Workshop on Language Resources for European Minority Languages*, Granada.
- ³²<http://nipadio.lsi.upc.es/cgi-bin/demo/demo.pl>
- ³³AHMAD. & DAVIES (1994). 'weirdness' in special-language text: Welsh radioactive chemicals texts as an exemplar. *Journal of the International Institute for Terminology Research*, 5(2), 22–52.

- ³⁴ ILARAZ ET AL. (2003). HIZKING21. In (Streiter *et al.*, 2003)
- ³⁵ <http://dev.eurac.edu/bistro>
- ³⁶ <http://www.termcat.es>
- ³⁷ http://www1.euskadi.net/euskalterm/indice_e.htm
- ³⁸ DAELEMANS ET AL. (2002). *TiMBL: Tilburg Memory-Based Learner, version 4.3*. Reference guide.
- ³⁹ <http://www.gnu.org/software>
- ⁴⁰ <http://odur.let.rug.nl/~vannoord/TextCat>
- ⁴¹ <http://dev.eurac.edu:8080/perl/all.tar.gz>
- ⁴² <http://dev.eurac.edu:8080/autoren/mitarbeiter/ostreiter/octopus.tar.gz>
- ⁴³ <http://www.buber.net/Basque; http://www.euskaracorpusa.net/XXmendea/index.html>
- ⁴⁴ <http://webs.uvigo.es/h06/weba573/personal/henr/recurs/bibl1.htm>
- ⁴⁵ <http://corpus.cirp.es/corga/info.html>
- ⁴⁶ <http://www.ling.lancs.ac.uk/staff/mark/cwbc/cwbcman.htm>
- ⁴⁷ <http://www.ucc.ie/celt/search.html>
- ⁴⁸ <http://www.smo.uhi.ac.uk/~oduibhin/tobar/>
- ⁴⁹ http://www.bangor.ac.uk/ar/cb/ceg/ceg_eng.html
- ⁵⁰ <http://www.spinfo.uni-koeln.de/mensch/textos.html; http://www.unionesarda.it/UNIONE/servizi/ricerche.htm>
- ⁵¹ <http://airas.cirp.es/WXN/wxn/frames/meddb.html>
- ⁵² <http://www.unhchr.ch/udhr/navigate/alpha.htm>
- ⁵³ <http://www.lrc.salemstate.edu/aske/basquecorpus/>
- ⁵⁴ <http://www.framtak.com/info/sounds.html>
- ⁵⁵ <http://www.byheart.freeservers.com/stuff.html>
- ⁵⁶ <http://tales.itc.it/WebDilf/servlets/index.html>
- ⁵⁷ <http://www.newcastle.edu.au/centre/amrhd/awaba/language/dictionary/index.html>
- ⁵⁸ <http://www.francenet.fr/~perrot/breizh>
- ⁵⁹ <http://www.kreyol.com/dictionary.html>
- ⁶⁰ <http://www.crannog.ie/focloir.htm>
- ⁶¹ <http://www.freelang.com/freelang/dictionnaire/afrikaans.html>
- ⁶² <ftp://storm.ptc.spbu.ru/pub/human-language/tibetan/t.arj>
- ⁶³ <http://www.cirp.es/lis/listas.html>
- ⁶⁴ <http://galego.org/english/dictionary.html>

References

- CARL M. & LANGLAIS P. (2003). Tuning general translation knowledge to a sublanguage. In *Proc. of CLAW 2003*, Dublin, Ireland.
- DAELEMANS W., VAN DEN BOSCH A. & ZAVREL J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning, special issue*, (34), 11–43.
- DAY D., ABERDEEN J., HIRSCHMAN L., KOZIEROK R., ROBINSON P. & VILAIN M. (1997). Mixed-initiative development of language processing systems. In *Fifth Conference on Applied Natural Language Processing*, Washington D.C.
- SCANNELL K. P. (2003). Automatic thesaurus generation for minority languages: an Irish example. In (Streiter *et al.*, 2003).
- SOMERS H. L. (1997). Machine translation and minority languages. In *Translating and the Computer 19, Aslib*, London.
- SOMERS H. L. (1998). Language resources and minority languages. *Language Today*, 5, 20–24.
- SOMERS H. L. (2001). EBMT seen as Case-based Reasoning. In *Proc. of the Workshop on Example-Based Machine Translation, at the MT-Summit*, Santiago de Compostela.

STREITER O. (2001). Corpus-based parsing and treebank development. In *ICCPOL 2001, 19th Intern. Conference on Computer Processing of Oriental Languages*, p. 115–120, Seoul.

STREITER O. (2002a). Abduction, induction and memorizing in corpus-based parsing. In *ESSLLI-2002 Workshop on "Machine Learning Approaches in Computational Linguistics"*, p. 73–90, Trento, Italy.

STREITER O. (2002b). Treebank development with deductive and abductive explanation-based learning: Exploratory experiments. In *Workshop on Treebanks and Linguistic Theories 2002*, Sozopol, Bulgaria.

STREITER O., ZIELINSKI D., TIES I. & VOLTMER L. (2002). Example-based term extraction for minority languages. In *Soziolinguistica y Language Planning*, Urtijëi/St. Ulrich/ Ortisei.

STREITER O., STUFLESSER M., & VOLTMER L. (EDS) (2003). *Proc. of Workshop on Natural Language Processing of Minority Languages with Few Computational Linguistic Resources*, Batz-sur-Mer, France.

A Language Index

Table 6: Short Description of Minority Languages cited, official languages and speakers

Key	Language	Language family	Speakers	Land/Region	Official language
Awa	Awabakal			Newcastle (AU)	English
Afr	Afrikaans	Germanic	7.000.000	South-Africa, Namibia	English
Bas	Basque	Basque	700.000 100.000	Basque Country (ES) France	Spanish, Basque French
Bre	Breton	Celtic	250.000	Bretagne (FR)	French
BSL	British Sign Language	Deaf sign language	400.000	Britain	English
Cat	Catalan	Romance	9.000.000 260.000 22.000	Catalunya (ES) France Sardegna (IT)	Spanish, Catalan French Italian
Far	Faroese	Germanic	40.000	Faroe Islands	Faroese
Fri	West Frisian	Germanic	350.000	Friesland (NL)	Dutch, Frisian
Gal	Galician	Romance	3.000.000	Galicia (ES)	Spanish, Galician
HCr	Haitian Creole		5.700.000 1.000.000	Haiti USA	French English
Iri	Irish	Celtic	400.000	Ireland	Irish, English
Lad	Ladin	Romance	17.000	Dolomites (IT)	Italian, German
Min	Mín Nán	Sino-Tibetan	15.000.000 26.000.000 1.000.000	Taiwan China Thailand	Mandarin Mandarin Thai
Sar	Sardinian	Romance	1.500.000	Sardegna (IT)	Italian
Tib	Tibetan	Sino-Tibetan	1.000.000 120.000 60.000	China India Nepal	Chinese Hindi etc. Nepali, Gurung
TSS	Taiwanese Sign Language	Deaf sign language	30.000	Taiwan	Mandarin
Wal	Walloon	Romance	600.000	Belgium	French
Wel	Welsh	Celtic	600.000	Wales	English

Workshop on NLP of Minority Languages and Small Languages TALN 2003

HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities

A. Diaz de Ilarza (1), A. Gurrutxaga (2), I. Hernaez (1),
N. Lopez de Gereñu (3) and K. Sarasola (1)

(1) Ixa taldea – University of the Basque Country
649 Postakutxa. 20080 Donostia

jipsagak@si.ehu.es

(2) Elhuyar Fundazioa

Astesuain Poligonoa, 14 - 20170 Usurbil

agurrutxaga@elhuyar.com

(3) VicomTech

Miramon Technology Park, 20009 Donostia

nlopez@vicomtech.es

Résumé – Abstract

We present the main lines of the HIZKING21 project. Its main objective is to promote basic research in language engineering, orienting this investigation towards the requirements of the globalized environment of the present day. Our scope of work is the development of language technologies for the Basque language, as well as the integration of resources and tools for the language industry, both already existing resources and resources to be developed in this project, into different devices (PCs, PDAs, electrical household appliances, car equipment and so on). Our goal is to contribute to the easy and user-friendly interaction with all kind of devices, using language as the natural means of communication. The starting up of this project has been possible thanks to the advances and developments in the Natural Language Processing and Language Engineering for Basque language made by the participants of this projects in the last fifteen years, as well as to the fact of sharing a vision of to the more adequate strategy for the development of these technologies in the case of a minority language like Basque.

Mots Clés – Keywords

Traitement automatique du Basque, corpus, applications
NLP for Basque, corpora, tools, applications

1 Introduction

The groups that make up the HIZKING21 project are aware of a double matter. On the one hand, the necessity of developing interfaces which will make possible the easy and intuitive interaction between humans and all kind of devices, no matter their technological complexity. The great importance of the language to fulfil this interaction is clear. On the other hand, they are also aware of the fact that people should not have to renounce to the use of their mother language to do so. At present, even if the amount of products in the language industries for English is quite impressive, those products do not have the same spreading in other languages. Taking into account that we live in a multilingual society in Europe, we have, as European researchers who work in this area, a special training to develop multilingual products. And, as Basque researchers, we have also some kind of commitment towards the integration of Basque language in the Information Society. Besides, Basque can be used to prove the adequacy of products to suit other languages, specially minority languages that suffer from the same kind of scarcity. A special attention has been paid to the design of the groups that participate in the project, combining a R+D group from university, a foundation working on the development of Basque language for a long time and two technological centres, so that we can concentrate our efforts on experiences in the areas of research, development and commercialisation.

HIZKING21 has been presented as a project for strategic research within the *Etortek* program of the Department of Industry, Trade and Tourism of the Government of the Basque Autonomous Community. The general budget of the project was of 7 million euros, and it has been approved in a third of its content within the *Etortek* program, which means an initial financing of 16%.

Our presentation will consist of the following sections: a) general vision of the problems that minority and small languages, and specially the Basque language, have to confront in the areas of Natural Language Processing and development of the language technologies in general; b) description of the strategy that we propose for the development of these technologies; c) departure point of the project with respect to the basic technologies, resources and tools available nowadays for our language; d) general objectives of HIZKING21; e) specific objectives of HIZKING21 for basic investigation, generation of resources, development of tools and design of prototypes and pre-applications.

2 NLP and minority languages

A language that seeks to survive in the modern information society requires language technology products. Human Language Technologies are making an essential contribution to the success of the information society, but most of the working applications are available only in English. Minority languages have to make a great effort to face this challenge (Petek, 2000) (Williams et al., 2001).

Language technology development for minority languages differs in several aspects from the development for widely used languages. This is mainly due to two reasons.

On the one hand, the size of the speakers' community is usually small. As a result, most of these languages have not enough specialized human resources, they lack in financial support,

and commercial profitability is, almost in all cases, a very difficult goal to reach. In other words, lesser-used languages have to face up to the scarcity of the resources and tools that could make possible this development at a reasonable and competitive rate.

On the other hand, there are language-specific problems, related to language typology. For a lesser-used language it is not always possible to use or to adopt the language technologies developed for other languages. This is especially relevant in rule-based approaches, but also in corpus-based approaches, because truly efficient exploitation of corpus demands annotation, and this process is in most cases based on rule-based procedures, like morphological and syntactic analysis. For example, romance languages like Galician, Catalan or Occitan can take advantage of NLP developments for French or Spanish, but these developments are not so applicable to some languages, for example Basque. This applicability (or portability) depends largely on language similarity. Basque is an agglutinative language, with a rich flexional morphology, and this requires specific procedures for language analysis and generation.

3 A strategy to develop language technology for a lesser used language

We present here an open proposal for making progress in Human Language Technology. Anyway, the steps here proposed do not correspond exactly with those observed in the history of the processing of English, because the high capacity and computational power of present computers allows to face problems in a different way.

Language foundations and research are essential to create any tool or application; but in the same way tools and applications will be very helpful in the research and improvement of language foundations. Therefore, these three levels (language foundations, tools and applications) have to be incrementally developed in a parallel and coordinated way in order to get the best benefit from them.

Our proposal is based on our experience with the automatic processing of Basque. Some features of Basque have to be known in order to evaluate the applicability of our strategy for other minority languages. As we have pointed out, Basque is an agglutinative language with a very rich morphology. It has basically constituent-free order at sentence level. There are around 700,000 Basque speakers, around 25% of the total population of the Basque Country, but they are not evenly distributed. There are six dialects, but since 1968 the Academy of the Basque Language (Euskaltzaindia) has been involved in a standardization process. At present, the morphology is completely standardized, but the lexical standardization process is underway.

We propose four phases as a general strategy for language processing.

3.1 Initial phase: Foundations

- Corpus I. Collection of raw text without any tagging mark.
- Lexical database I. The first version could be a list of lemmas and affixes.
- Machine-readable dictionaries.

- Morphological description.
- Speech corpus I.
- Description of phonemes.

3.2 Second phase: Basic tools and applications.

- Statistical tools for the treatment of corpus.
- Morphological analyzer/generator.
- Lemmatizer/tagger.
- Spelling checker and corrector (although in morphologically simple languages a word list could be enough).
- Speech processing at word level.
- Corpus II. Word-forms are tagged with their part of speech and lemma.
- Lexical database II. Lexical support for the construction of general applications, including part of speech and morphological information.

3.3 Third phase: Advanced tools and applications.

- An environment for tool integration. For example, following the lines defined by TEI using XML (Artola et al.; 2000).
- Web crawler. A traditional search machine that integrates lemmatization and language identification.
- Surface syntax.
- Corpus III. Syntactically tagged text.
- Grammar and style checkers.
- Structured versions of dictionaries. They allow enhanced functionality not available for printed or raw electronic versions.

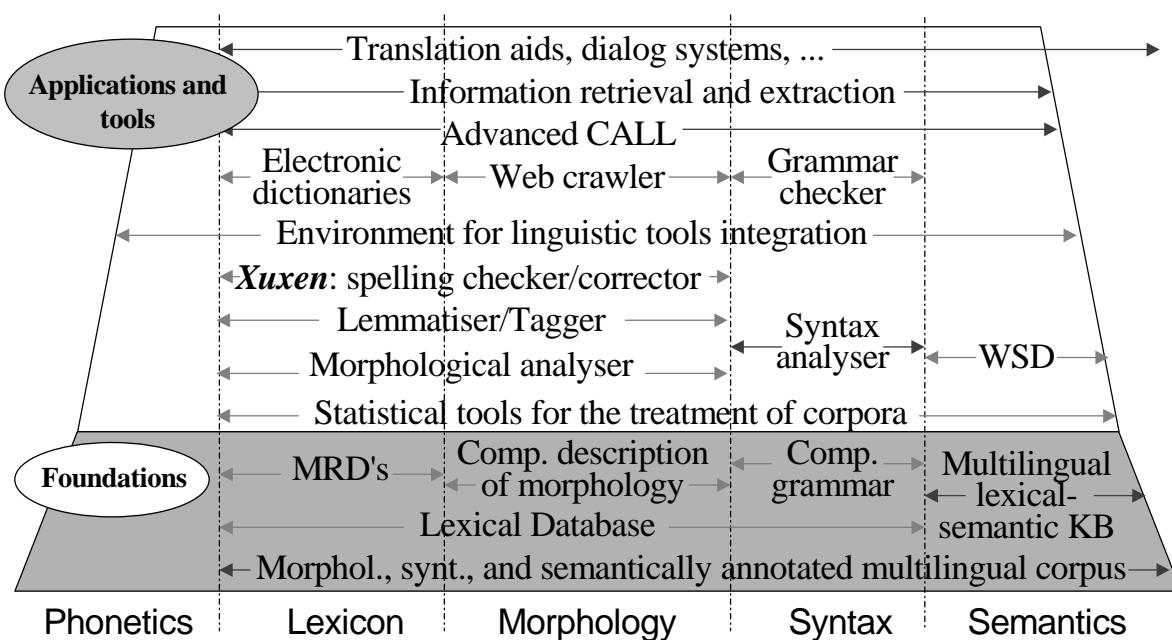


Figure 1: Foundations, applications and tools in language technologies development.

- Lexical database III. The previous version is enriched with multiword lexical units.
- Integration of dictionaries in text editors.
- Lexical-semantic knowledge base. Creation of a concept taxonomy (e.g.: Wordnet).
- Word-sense disambiguation.
- Speech processing at sentence level.
- Basic Computer Aided Language Learning (CALL) systems.

3.4 Fourth phase: Multilingualism and general applications.

- Information retrieval and extraction.
- Translation aids. Integrated use of multiple on-line dictionaries, translation of noun phrases and simple sentences.
- Corpus IV. Semantically tagged text after word-sense disambiguation.
- Dialog systems.
- Knowledge base on multilingual lexico-semantic relations and its applications.

We will complete this strategy with some suggestions about what shouldn't be done when working on the treatment of minority languages. a) Do not start developing applications if linguistic foundations are not defined previously; we recommend to follow the above given order: foundations, tools and applications. b) When a new system has to be planned, do not create ad hoc lexical or syntactic resources; you should design those resources in a way that they could be easily extended to full coverage and reusable by any other tool or application. c) If you complete a new resource or tool, do not keep it to yourself; there are many researchers working on English, but only a few on each minority language; thus, the few results should be public and shared for research purposes, for it is desirable to avoid needless and costly repetition of work.

4 Technologies, resources and tools available nowadays for Basque

It is well known that in the last fifteen years several research groups in the Basque Country have been working on this field with the common aim of developing basic computational resources and tools for Basque. The leaders of this work have been two groups of the University of the Basque Country:

- The Aholab group (bips.bi.ehu.es/ahoweb), specialized in speech technologies (synthesis and recognition); it belongs to the Signal Treatment and Radiocommunication Team of Electronics and Telecommunication Department of the University of the Basque Country
- The IXA group (ixa.si.ehu.es), specialized in the processing of written texts at different levels (morphology, syntax, semantics; corpora, machine translation, IE_IR...); the group is made up mainly of members of the Computer Science Faculty of the University of the Basque Country (computer scientists and linguists), and by members of UZEI (Basque Center for Terminology & Lexicography)

Nowadays, the most remarkable tools ready for use are:

- A lemmatizer/tagger (EUSLEM) developed by the IXA group in collaboration with UZEI, and based on a two-level morphological analyzer (MORFEUS) and a lexical database (EDBL), and improved recently with disambiguation module based on the Constraint Grammar formalism and a module that treats Multiword Lexical Units (Ezeiza et al., 1998).
- Basque WordNet (BasWN, ixa.si.ehu.es/Ixa/PilPilean/1022169813), implemented for Basque by the IXA group (Agirre et al., 2002)
- The speech synthesizer developed by Aholab (Navas et al., 2002).

Several applications have been commercialized using these tools:

- A spelling checker (XUXEN)
- A lemmatization based web-crawler (GAIN)
- A lemmatization based on-line bilingual dictionary (*Elhuyar Hiztegi Txikia-ren plugin-a Microsoft Word 2000*; "Basque-Spanish/Spanish-Basque Small Elhuyar Dictionary plug-in for Microsoft Word 2000")
- A generator of weather reports (MultiMeteo)
- A text-to-speech application (AhoTTS).

With regard to corpus resources, there are nowadays two significant corpora of Basque texts:

- The textual corpus of the *OEH-Orotariko Euskal Hiztegia* ("Basque General Dictionary"): a non-lemmatized corpus that collects all of Basque written texts until language standardization (~1960). It has approximately 5,5 million words.
- The *XX. mendeko Euskararen Corpus Estatistikoa* ("The statistical corpus of 20th Century Basque"): a lemmatized corpus with feature-structure markup in SGML, implemented on the ORACLE relational database; it can be consulted on line (http://www.euskaracorpusa.net/XXmendea/Konts_arrunta_fr.html). Its size is of 4.658.036 words.

In recent years, several private companies and technology centers of the Basque Country have begun to get interested and to invest in this area. At the same time, more agents have come to be aware of the fact that collaboration is essential to the development of language technologies for minority languages. One of the fruits of this collaboration is the HIZKING21 project. Together with the IXA and Aholab groups mentioned above, the followings organizations take part in HIZKING21:

- Vicomtech: an applied research center (www.vicomtech.es) working in the area of interactive computer graphics and digital multimedia. It was founded jointly by the INI-GraphicsNet Foundation and by EiTB, the Basque Radio and Television Group.
- Elhuyar Foundation: a non-profit organization (www.elhuyar.com) aimed to promote the normalization and standardization of Basque, with activities in the fields of lexicography and terminology, dictionary publishing, language planning, science and technology communication, textbooks and multimedia products and services
- Robotiker: a technology center (www.robotiker.com) specialized in Information and Telecommunication technologies. Robotiker is part of Tecnalia Technology Corporation.

5 HIZKING21: general objectives

Starting off the previously described situation, the HIZKING21 promoters consider necessary: a) to combine and coordinate efforts and to share knowledge and resources in order to deepen, accelerate and spread the developments and profits in the area of the technology of the Basque language; b) to design and implement the prototypes and pre-applications that will make possible the development, in the medium term, of commercial products: different systems with linguistic capacity in Basque language. The conviction that we share about these needs has been the base of our motivation to create the HIZKING21 project.

We propose the following general objectives for HIZKING21:

- To design a common strategy and to create a network of excellence in R+D in the area of language technologies, in accordance with international trends through strategic alliances with centers of reference and projects
- To provide the Basque language in the 2006 with a similar level of resources and tools that those available at present for English, in the scope of speech technologies and computational linguistics
- To promote the exploitation of the opportunities offered by language technologies in:
 - Content management
 - Recognition and synthesis of voice
 - Document production and translation technology
 - E-learning
 - Multimedia systems
- To lay the foundations that will facilitate the internationalization of our linguistic industry in the very near future

The main goal of the project is the development of what we call "**systems with linguistic capabilities**", which will have to be multilingual and guarantee multimodal interaction with users. To do so, the whole project has been divided into different tasks.

- First of all, it is necessary to *identify all of the components that make up such a system*: multimedia interfaces for each environment, resources and tools needed for each kind of system, level of development and availability of existing components, and so on. The main purpose of this stage is to get a clear image about which are the necessary components already available; the ones to be developed from scratch in Basque Country, either in existing centres or in centres to be created; or components that, existing for other languages, could be adapted to Basque. This clear image will make possible the definition of more specific steps to follow during subsequent stages of the project.
- Second stage will be the development of all the works defined in the first stage. It will be necessary to arrange them into different groups, according to both the mentioned strategy in the processing of the language and to the technological lines involved. The technical groups defined are: Corpus, Resources, Tools, Basic Technologies and Pre-applications. These technical groups are to work in a coordinated way in order to achieve one of the goals of HIZKING21: the design of *devices based on user-friendly and easy interaction through the use of language*,

- offering an important contribution to the spread of the use of Information Society Technologies by EVERYONE though the reduction of complexity in their use.
- Another important task to develop in HIZKING21 all along the project is to promote qualification and high level training of people in the scope of language technologies, in order to increase research ability in Basque Country. To do so, exchanges off people with important centres of reference, universities and companies around Europe are planned, together with postgraduate courses and the realization of doctoral thesis. It will be necessary to reach collaboration agreements with these mentioned centres.
- The sharing of knowledge is another critical element in the achievement of the goals of HIZKING21, due to the great necessity of taking advantage of all synergies that may arise among different communities of researchers working on minority languages. So technological surveillance and dissemination will concentrate important efforts of the people involved in the project. Implementation of a systematized method of surveillance, demonstration meetings, publication of results in specialized journals and conferences, thematic seminars and courses, technological meetings, and so on, are some of the activities planned. There will also be a web site where to publish all of the relevant information related to HIZKING21, and to become important information exchange point during the whole development of the project.

6 HIZKING21: detailed objectives in NLP

Within HIZKING21 project,, the most important areas are the following: R+D, training, infrastructure, international collaboration, diffusion and the creation of a technological observatory. We have fixed concrete objectives for basic investigation, generation of resources, development of tools and design of prototypes and pre-applications. The nucleus of the development HIZKING21 is the accomplishment of research and development projects. The following are initially considered as high priority projects:

- Development of basic linguistic resources
- Development of basic tools
- Technical and basic methods for integration of resources and tools
- Person/Machine Interfaces and their integration in multimedia environments
- Wireless technology and hardware associated to the systems with linguistic capacity

The first two aims of them are described next.

6.1 Development of basic linguistic resources

A set of corpora, lexical databases and electronic dictionaries will be completed or created during the development of the project.

6.1.1 *Written corpora:*

- Corpus I. Collection of raw text without any tagging mark (light XML)

- Corpus IIa. Word-forms are tagged with lemma, POS and morphosyntax analysis (hand corrected)
- Corpus III. Syntactically tagged text (hand corrected)
- Corpus IV. Semantically tagged text after word-sense disambiguation (hand corrected)
- Corpus Va. Multilingual corpora (light XML)
- Corpus Vb. Multilingual and parallel corpora

6.1.2 *Spoken corpora:*

- Development of text corpus for tasks of Automatic Speech Recognition (ASR)
- Creation of phonetic voice corpus for tasks to 16KHz for all the dialectal varieties of Basque

6.1.3 *Lexical databases*

- Lexical database with information about POS, syntax information, multiword units, verb subcategorization and collocations
- Lexical database with semantic information linked to ontologies
- Lexico-semantic database. Concept taxonomy
- Multilingual lexico-semantic database. Concept taxonomy

6.1.4 *Electronic dictionaries*

- Integration in a data bank of lexicographical and terminological databases
- Design and implementation of a lexicographic workbench
- Design and implementation of a terminological workbench

6.2 Development of basic tools

The next tools will be improved or generated in the project:

- Speech processing: Large Vocabulary Continuous Speech Recognition (LVCSR), development of a high-level TTS system, ASR system
- Syntax: identification of multiword units, definition of syntactic mark-up, syntax analyzer
- Semantics: document classification, entity identification and processing, word-sense disambiguation
- Pragmatics and Discourse: resolution of pronominal anaphora, definition of the structure, goals and topics of a dialog system
- Corpus analysis tools:
- Information retrieval from corpus marked up in XML (concordances, statistics, collocations...)
- Linguistics and statistical tools for terminology extraction from tagged corpora
- Linguistics and statistical tools for the extraction of lexical and terminological equivalences from multilingual tagged corpora

Acknowledgments

This project is partially supported by the *Etortek* program of the Department of Industry, Trade and Tourism of the local Government of the Basque Country.

References

- Agirre E., Ansa O., Arregi X., Arriola J., Díaz de Ilarraz A., Pociello E., Uria L. (2002) Methodological issues in the building of the Basque WordNet: quantitative and qualitative analysis. Proceedings of First International WordNet Conference. Mysore (India).
- Artola X., Díaz de Ilarraz A., Ezeiza N., Gojenola K., Maritxalar M., Soroa A. (2000), A proposal for The Integration of NLP Tools using SGML-Tagged documents, Second Int. Conf. on Language Resources and Evaluation. Athens (Greece). May, 2000
- Ezeiza N., Aduriz I., Alegria I., Arriola J.M., Urizar R. (1998), Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages, *COLING-ACL'98*, Montreal (Canada). August 10-14, 1998.
- Navas E., Hernaez I., Ezeiza N. (2002) Assigning Phrase Breaks using CART's in Basque TTS. Proc. of the 1st Int. Conf. on Speech Prosody, Aix-en-Provence, pp. 527-531, 2002
- Petek B. (2000), Funding for research into human language technologies for less prevalent languages, *Second International Conference on Language Resources and Evaluation (LREC 2000)*. Athens, Greece.
- Williams B., Sarasola K., Ó'Croinin D., Petek B. (2001), Speech and Language Technology for Minority Languages. *Proceedings of Eurospeech 2001*

TALN 2003
Linguistic Resources and Infrastructures for the Automatic
Treatment of Ladin Language

Evelyn Bortolotti (1) and Sabrina Rasom (2)

Istitut Cultural Ladin “majon di fascegn” (ICL)
Via della Chiesa 6 38039 Vigo di Fassa (TN) - Italia
(1) rep.ling@istladin.net
(2) lengaz@istladin.net

ITC-IRST
Trento – Italia

Abstract - Résumé

In this paper we present the electronic infrastructures created for the automatic treatment of the Central Ladin language: the tool-box for lexicographers and the spell-checker of Fassan Ladin. We focus our attention particularly on the development of these tools, on their application and on their limits, presenting the work of creation of corpora and their organisation in thematic folders. The tool-box is a very useful instrument for linguistic researches and it allows the approach to language in a most modern way. It is a tool that can and will be implemented and improved according to the needs of the user.

As for the spell-checker we describe all the work of elaboration of the morphological rules, the actual organisation of these rules in order to solve spelling and accent problems and the process followed in generating the form list of Fassan Ladin. Furthermore we present the possible rectifications it can do, the limits of this first version, and the solutions we are going to adopt in order to improve this tool.

Keywords – Mots Clés

French Key Words.

Analyse morphologique, analyse syntaxique, banques de données, concordances, corpus, corpus planning, dictionnaire électronique, étiquetage, langues minoritaires, linguistique computationale, lexicographie, spell-checker, standardisation, terminologie, tool-box, traitement automatique.

English Key Words.

Automatic treatment, concordances, corpus, corpus planning, computational lexicography, databases, electronic dictionary, lexicography, minority languages, morphological analysis, spell-checker, standardisation, syntactical analysis, tagging, terminology, tool-box.

Introduction

The infrastructures for the automatic treatment of Ladin language we have been developing up to now comprehend a tool-box for lexicographers and a beta version of the spell-checker of Fassan Ladin Language. The tool-box for lexicographers is a web application thanks to which the user can have free and easy access to a wide range of lexical resources by using any web browser, thus being able to easily consult and modify lexical data. Different kinds of resources can be consulted at the moment: the lexical databases that had been collected within the project SPELL, the textual corpora of the different Dolomite Ladin varieties, and the modern normative bilingual dictionaries that have been created by the Ladin institutes. The tool-box contains a range of different tasks, which we are going to analyse in this paper. In Section 1 we will describe the way in which we created textual corpora (1.1), the tool-box and its tasks (1.2), its application fields and its limits (1.3).

The spell-checker is a tool, which allows people approaching to written Fassan Ladin to write in a simpler, and easier way according to the spelling rules created in order to have a unified language. The introduction of Ladin in the public administration, in the schools and in the media brought to a new demand of tools which could offer to people working with this language a range of useful tasks. A first step in this direction is the creation of the spell-checker of Fassan Ladin. In Section 2 we will present the ways followed in realising a first version of the spell-checker: the elaboration of the rules (2.1), the development of its tasks (2.2), its limits (2.3), and the future intents.

1 The Tool-box¹

The development of new infrastructures for the treatment of Ladin Language aims at providing ICL (Istitut Cultural Ladin “Majon di Fascegn”) and the other Ladin cultural institutions with technological tools and infrastructures that could improve and optimise linguistic studies and elaboration. Because of the need to simultaneously work on great corpora and on several dictionaries, the software previously used by the linguists of the project had become inadequate. First of all, we had to use different applications up to now, even when carrying on the same work on different lexical resources. For example, in order to search the same word in different dictionaries, we had to repeat the research in each of them. Second, the lexical resources were in different formats, which moreover were often inadequate for the used applications. Last, whereas the corpora are rather fixed, the dictionaries keep changing, so the work of distributing updated copies to the project partners could be complex and expensive, especially when more linguists worked on the same resource. These conditions brought us to establish two main goals: the conversion of the existing lexical resources into a standard format, and the

¹ The Tool-box for lexicographers has been developed inside the European project TALES (Linguistic Resources and Infrastructures for the Automatic Treatment of Ladin and Sard), which began in November 1999 and is still going on. This project is carried on by the linguistic staff of Ladin Culture Institute “majon di fascegn” (ICL) of Fassa Valley and Dr. Claudio Giuliano of ITC-IRST (Institute for Scientific and Technological Research) of Trento.

development of an application that could allow us to consult and modify linguistic data through a unique and uniform interface.

Several applications have been recently developed in order to access and modify linguistic data, but the difficulties in adjusting these tools to our project have required the re-implementation of a new application. Thus, the tool-box is a custom-made system, which assembles many characteristics and functions of other systems.

1.1 Creation of textual corpora

One of the first steps in the establishment of textual corpora regarded the creation of machine-readable corpora of texts representing the different idioms belonging to Dolomite Ladin. Just few texts were already available as data processing files. We had therefore to increase their number both through automatic acquisition of texts, i.e. particularly by scanning recent texts, and through manual digitising, first of all of older texts. The latter furthermore demanded a careful work of spelling normalisation, as many phonemes were once rendered through a huge number of diacritics that could not be read by the machines.

In the meantime we also went on classifying the corpora by dividing the texts according to different criteria. The division regards three aspects: diatopic variation, diachrony, and textual typology.

We determined the following genres:

Literary texts: author literature, folk literature (tales, legends, anecdotes), theatre, poetry, folklore (usage and traditions), memoir writing, paroemiology, religion.

Non literary texts: administration, law, economics, popular scientific and technological writings, popular cultural writings, newspaper information, political writings, pragmatic writings, school writings.

Each text was also saved in a .txt file and put into different folders corresponding to the genres described. Then we created a file of Excel containing a header for each file, which provides all the important information about the texts, i.e.: the time they were originally written, the author, the information about the folder they are stored in, the file name, the original title of the text, their source (the book or magazine from which they had been taken out), the number of words, the sub-variety and whether they have been graphically standardized.

	L1	Casella Nome	B	C	D	E	F	G	H	I	J	K	
			DATA	AUTOR	PERCORS	INOM	TITOL	FONTE	EDITOR	SORT LETTERARA	n. PAROLE	VARIA NTE	GRAFIA
1	1977	Vito Chiocchetti da Vigo	corpus:divulgaz_culturale:MONDO_LADINO_77		A1_teater_da_Vich	A1 teater da Vich	ML77	ICL Majon di Fasegn		divulgazion e culturale	607	brach	NORM
2	1977	Vito Chiocchetti da Vigo	corpus:teatro:MONDO_LADINO:Vito_Chiocchetti:Teater_da_Vich		Son_chiò_par_colpa_voscia	Son chiò par colpa voscia	ML77	ICL Majon di Fasegn		teatro	833	brach	NORM
3	1977	Vito Chiocchetti da Vigo	corpus:teatro:MONDO_LADINO:Vito_Chiocchetti:Teater_da_Vich		I_sposc_e_la_bastia	I sposc e la bastia	ML77	ICL Majon di Fasegn		teatro	944	brach	NORM
4	1977	Vito Chiocchetti da Vigo	corpus:teatro:MONDO_LADINO:Vito_Chiocchetti:Teater_da_Vich		El_guerier_soldà	El guerier soldà e la grana de Sen Jan	ML77	ICL Majon di Fasegn		teatro	1062	brach	NORM
5	1978	Maria e Rosa Chiocchetti Menghie	corpus:folclore:MONDO_LA DINO:Maria e Rosa Menghie		ML78II-Canche_fajeane_lesciva	Canche fajeane lesciva	ML78 2	ICL Majon di Fasegn		folclore	623	moenat	NORM
6	1979	Ermanno Badia Pescol	corpus:teatro:MONDO_LADINO:Ermanno_Badia:ML79_3-4_Mascherèdes_da_chi		Janagnol_da_Penia_ch he_ven_ju	Janagnol da Penia che venju la val de sot a manidèr via la fia	ML79 3-4	ICL Majon di Fasegn		teatro	496	cazet	NORM
7	1979	Ermanno Badia Pescol	corpus:teatro:MONDO_LADINO:Ermanno_Badia:ML79_3-4_Mascherèdes_da_chi		Mascherada_par_dial et_fascian	Mascherada par dialet fascian	ML79 3-4	ICL Majon di Fasegn		teatro	1153	1049 cazet 104 brach	NORM
8	1979	Ermanno Badia Pescol	corpus:teatro:MONDO_LADINO:Ermanno_Badia:ML79_3-4_Mascherèdes_da_chi		Mascherèdes_III	Mascherada III	ML79 3-4	ICL Majon di Fasegn		teatro	1203	cazet	NORM
9	1980	Otavio Doliana da Fera	corpus:paresiologica:MONDO_LADINO:Otavio_Dolian_a_proverbs		ML3-4_1980_Proverbs	Proverbs	ML80 3-4	ICL Majon di Fasegn	paresiologi ca		1500	brach	NORM

Figure 1: the file of Excel with path

The results of the elaboration and division of these texts were then used for the creation of the tool-box for lexicographers. Thanks to the folders properly divided into different genres and the files with the path information, it was possible to create a range of useful tasks for linguistic inquiries about Ladin varieties.

1.2 The tool-box and its tasks

The tool-box² offers different tasks for the study of the texts. It is possible to analyse **concordances, collocations and frequencies** of words. It can be used for morphological, syntactical and lexical researches. The words can be searched in different ways according to the intention of the user. The resulting word is shown in a brief context and is written in red in order to be easily recognised. Also the words coming before and after it can be evidenced in a different colour (green) and the search interface allows the user to decide how many words to evidence. On the left hand of the interface you can find some information about the source from which the text was taken. By clicking on the source you open a new window with a longer context. On the top of the new window there is the complete information path. In order to make the research faster all the occurrences found are distributed on several pages. In the case of the frequencies the pages are also referred to with alphabetical letters. The syntax used to find the words is similar to that of the common research engines.

² The tool is available online at <http://tales.itc.it/resources.html>.

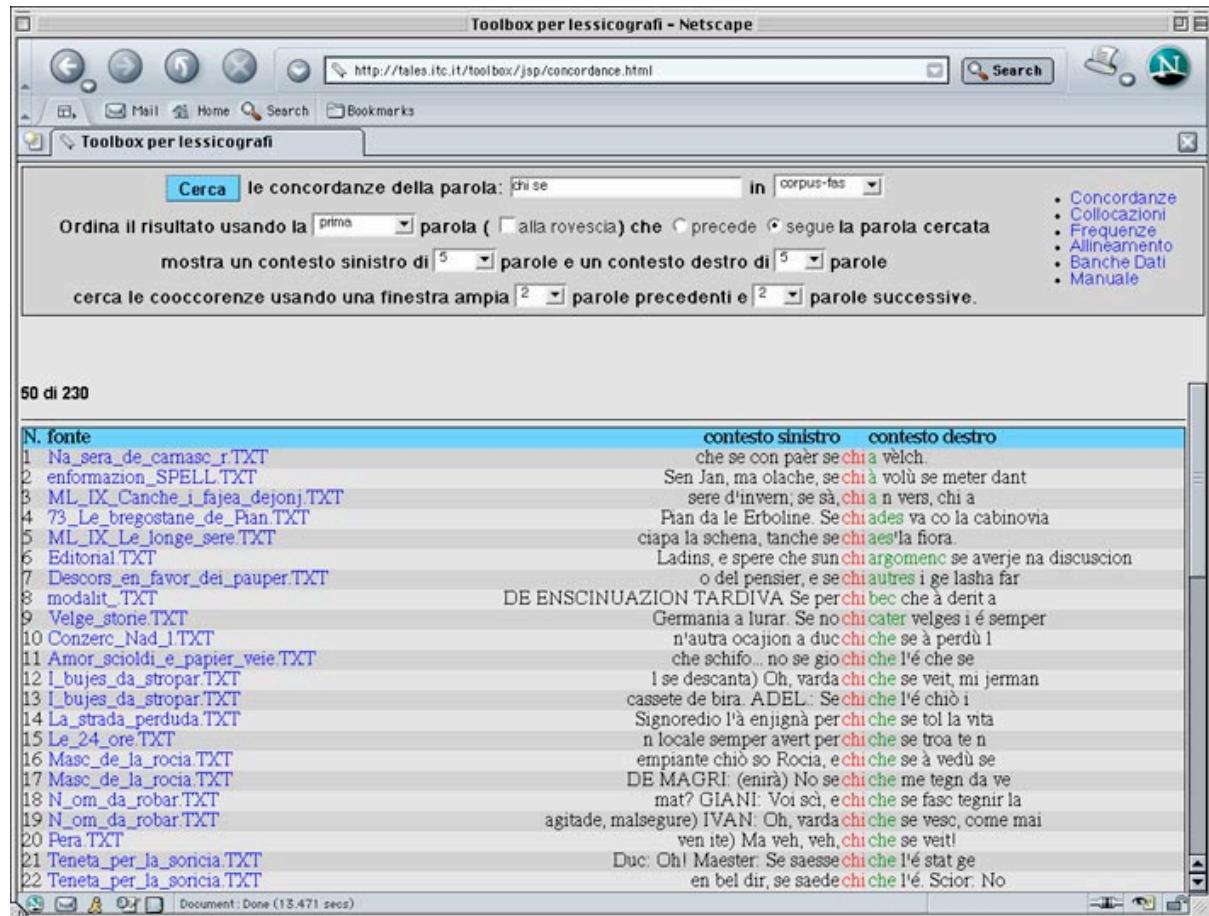


Figure 2: A concordance research

A particular task has been created for the consultation and elaboration of **administrative texts**. This corpus is a parallel one. Administrative writings are bilingual (Fassan–Italian) or trilingual (Badiot/Gherdëina – Italian – German). We gathered all the original writings from the public offices and saved them in a file .doc. Then, in order to create a task useful for the consultation and for a parallel research of the words or strings of words, each text was divided in small paragraphs were it could be possible to recognise the correspondent (or parallel) words in the other language or languages. These files thus divided were also further elaborated and used by EURAC for creating a program for administrative term extraction.

LA GIUNTA COMPRENSORIALE:	LA JONTA COMPRENJORIALA:
<p>- richiamate le deliberazioni giuntali n. 93 del 19 marzo 1997 e n. 170 del 14.05.1997, con le quali si impegnava la spesa di lire 82.252.000.= per gli inserimenti in forma residenziale di due utenti di Campitello di Fassa presso l'Istituto "Cooperativa Sociale Villa Maria" di Lenzima di Isera e l'"Istituto Ospedaliero di Sospiro" di Sospiro (Cr) durante l'anno 1997;</p>	<p>- tout cà la deliberazions de Jonta nr 93 dai 19 de mèrz 1997 e nr 170 dai 14.05.1997, con cheles che vegnia empegnà na speisa de lires 82.252.000.= per i enseriment a na moda residenziala de doi utenc de Ciampedel aló da l'Istitut "Coprativa Soziala Villa Maria" de Lenzima de Isera e l'"Istitut Ospedalier d Sospiro" de Sospiro (Cr) via per l'an 1997;</p>
<p>- preso atto della comunicazione P.A.T. prot. n. 995/ASS/52/MLM/mlm del 25 luglio 1997, da cui risulta che la retta giornaliera applicata dall'Istituto "Cooperativa Sociale Villa Maria" è di lire 180.000.= per l'anno 1997;</p>	<p>- tout at de la comunicazion P.A.T. prot. nr 995/ASS/52/MLM/mlm dai 25 de messèl 1997, da olache resultea che la quota per di fata da l'Istitut "Cooperativa Sociale Villa Maria" la é de lires 180.000.= per l'an 1997;</p>

Figure 3: An example of parallel text

Another task of the tool-box comprehends the **lexical databases** of all varieties (Fascian, Badiot, Gherdeïna, Anpezan, and Fodom). The original databases in HyperCard or File Maker have been made compatible in order to be consulted through the interface of the tool-box. Thanks to the databases it is possible to consult the thesaurus belonging to each variety finding meanings, phraseology and sources in a variety or in all the varieties together, according to the actual needs of the user.

A last application available in the tool-box is the consultation of the **modern bilingual dictionaries** (DILF³ for Fassan Ladin, Mischi for Badiot and Forni for Gherdëina) edited by the Ladin institutes.

³ The DILF is available online at <http://tales.itc.it/WebDilf/servlets/index.html>.

Lemma	Abstract definizione
tavolo	sm. desch, -sf...
tavola	sf. 1. (asse, ripiano) brea, bree; (asse) pianicia, -ces...
ribalta	sf. 1. rebalza, -es...
togliere	v.t. 1. tor dernez, tol, tout; tor via; tor fora; tor jù; tor...
traballare	v.i. schiancolèr, schiàncola; centenèr, -ea; balèr, bala; scorlèr, scoria...

Figure 4: A concordance research

1.3 Application fields and limits of the tool-box

The tool-box has already been widely used as a most helpful instrument for the whole linguistic work of ICL. It was particularly useful both in the range of the lexicographical work and as support to the elaboration of the Fassan Ladin Grammar (*Gramatica del Ladin Fascian*) and of the materials for the alphabetisation courses (*Cors de alfabetisazion per ladinofons e Cors de alfabetisazion per no ladins*).

In particular, this system can be usefully consulted in order to analyse the Fassan corpus, which is up to now made up of about 2,000,000 words, thus strongly facilitating the study of lexicon, of syntax, and of morphology of Ladin and its local varieties. Furthermore, the analysis of textual corpora has a great importance in the work of corpus planning and standardisation of language, as it allows to verify the elaboration of rules and to give real usage examples.

Moreover, the tool-box can be successfully used within the work of development and elaboration of the lexical database of Ladin Fassan. This will take shape first of all in the realisation of VoLF (Vocabolèr del Ladin Fascian), a new lexicographical thesaurus which will collect all local varieties and the whole ancient and modern written production. The parallel corpora can be very helpful while working on multilingual terminology, as we have been doing in the project TermLeS (Lexical and Terminological Standardisation for Ladin and Sard), particularly with respect to administrative and environmental terminology.

The tool-box can be implemented and modified. First of all it is possible to add other texts. To do this we need to index the whole corpus again after adding the new files.

As for the limits of these tools, there are some aspects of the tasks that have to be improved and modified. The better way to understand how to implement the tool-box is to use it. The most important limits we have noticed up to now and that we are going to solve with the technical aid of IRST comprehend new ways to do restricted researches of words and the intent to render the tool faster and lighter. For the first intent we need to work out a path where we indicate the different filters we would like to use, i.e.: for example, it could be interesting to look for some words only in one author or in one sub variety. Thanks to these paths it will be possible to add this new task. On the other hand, one aspect we would like to improve is the task of the frequencies. As a matter of fact, it is not possible to look for the frequency of a word without consulting a static page containing all the frequencies of the words of the corpus up to now.

2 The spell-checker of Fassan Ladin

2.1 The development of the spell-checker of Fassan Ladin

In order to create this tool we first elaborated grammatical rules in files .xls with the morphology of the Fassan idiom on the base of the instructions given from the experts of the IRST, who should then use these files in order to create the form list of this variety. For every declinable grammatical class (nouns, adjectives, verbs) we gave the regular endings. Then we also listed all the exceptions, the only singular or only plural adjectives and nouns, and the invariable parts of the speech. For the verbs we listed all the persons and the times, and all the possible combinations between verb and object pronouns. We paid particular attention to the accents. In our idiom they depend on the division into syllables and that's way the addition of an ending can cause the need of an accent. In order to work out this problem we created a range of morphological adjustment rules. This kind of rules allows to solve spelling conventions without repeating or complicating the regular declinations. We tested these rules by applying them to all the entries of the dictionary DILF that was the lexical base for the creation of a first version of our form list. The rules thus applied gave results that could be compared with the information of the dictionary. If the forms generated through the rules were different, the files .xls had to be implemented and improved. These rules were afterwards used by IRST to create a form list. Finally the company Expert System of Modena realised the software.

In the process of elaboration of the morphological rules of the Fassan Ladin we also happened to find some inconsistencies regarding the standardisation rules already created and applied in the dictionary and in the grammar. These inconsistencies regarded particularly orthographic rules that resulted incomplete. Therefore, while elaborating these rules we could also examine closely the rules already existent and improve them.

	C	D	E	F	G	H
681 //		i sostantivi che al singolare hanno desinenza -l non preceduta da vocale -e presentano desinenza -i al plurale.				
682 def	sm-l	sm		i		
683 root	sm-l	.*[aèiouù]				
684						
685 example	sm-l	begin				
686	gamba-l	gamba-i				
687	segnè-l	segnè-i				
688	sti-l	sti-i				
689	sìmbo-l	sìmbo-i				
690						
691 end						
692						
693 exception	sm-l	begin				
694	pordeil	pordei				
695	cheil	cheiles				
696	peil	pei				
697	coul	coi				
698	crumsnobl	crumsnoboi				
699						
700 end						

Figure 5: Morphological rules

//		tutte le forme interrogative del modo indicativo che hanno desinenza -este, el, ela, ei, eles, hanno l'accento sull'ultima sillaba della radice se questa contiene vocale a, e, i, o, u.				
def vb	accenti-e	reg:ipi	.*	e [^aiou]* este eles ela ei el	é both	
example			begin			
		men-este	mén-este			
		men-el	mén-el			
		end				

Figure 6: Morphological adjustment rules

2.2 Functions of the checker

Now to the functions of the checker. It aims first at correcting a text while writing and then at correcting a text already written applying the checker in a second time. At the moment it is possible to rectify typing in and spelling mistakes, deviations from the standard Fassan (it is to say deviations from the standard idiom created in order to have a unified written language), morphological errors and uncorrected accents.

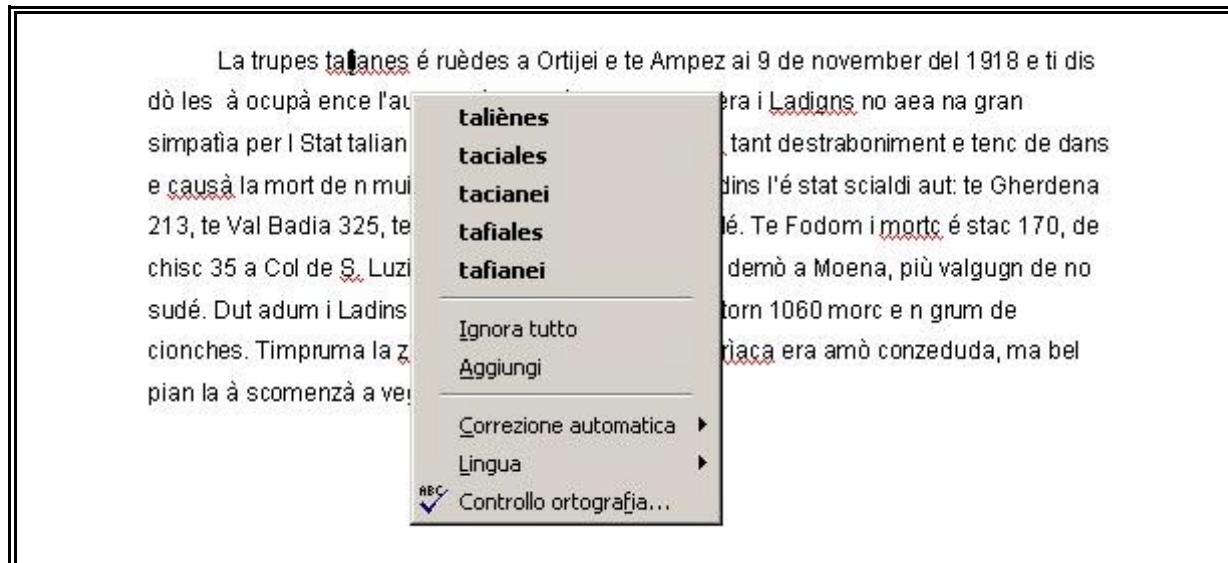


Figure 7: Example of spell checking

2.3 Limits of the checker and future intents

Because of the fact that this is a beta version of the checker it also has some limits to be considered. First of all it has a limited dictionary: in order to create our checker we based on the Fassan Ladin dictionary DILF (Dizionèr del Ladin Fascian) which, like all dictionaries, has a limited number of entries. For example it can lack traditional entries or sectorial terminology. Also the options suggested by the checker when a word is incorrect can be limited: maybe we do not find the expected correction. There is also the possibility of software flaws.

The project of development of the checker now consists first of all in improving and implementing the dictionary with lexicon taken from the corpora, with traditional entries and on the base of the personal dictionaries created by the people testing the tool. Then we are going to improve the wordformlist by checking and implementing the morphological rules, and to create new data processing rules, semantic fields and an automatic checker, in order to improve the suggestions given in the options. As for the semantic fields, it would be a very interesting work that could be used also in the case of creation of tagged texts.

In order to create a definitive version of the checker we decided to let people using the Fassan Ladin language every day for their work (for example teachers, employees and journalists) test this tool. They are asked to write up all the errors and the incongruities they find in using the checker and to create a personal dictionary where they will store all the words they use every day and which are not contained in the checker dictionary. We gave out the tool at the end of March and we are going to get back the personal dictionaries and the notes by the end of this year 2003. To inform people interested in using the spell-checker we organised a conference where we explained how to use the checker, how to create the personal dictionary and how to make suggestions to implement the tool.

Acknowledgments

We would like to thank Dr. Claudio Giuliano (ITC – IRST) for his suggestions.

Références - Bibliography

AA.VV. (2001), *Cors de alfabetisazion per ladinofons*, Vich, Comprenjorie Ladin de Fascia – Istitut Cultural Ladin.

AA.VV. (2001), *Cors de alfabetisazion no ladins*, Vich, Comprenjorie Ladin de Fascia – Istitut Cultural Ladin.

Chiocchetti N., Iori V. (2002), *Gramatica del Ladin Fascian*, Vich, Istitut Cultural Ladin.

DILF (2001 – 2002), *Dizionario Italiano – Ladino Fassano Dizionèr Talian – Ladin Fascian*, Vich, Istitut Cultural Ladin - SPELL.

Forni M. (2002), *Wörterbuch Deutsch – Grödner-Ladinisch Vocabuler Tudësch – Ladin de Gherdëina*, San Martin de Tor, Istitut Ladin “Micurà de Rü”.

Giuliano C. (2002), A tool box for lexicographers. In Proc. of EURALEX 2002, Conference Copenhagen , Denmark.

Mischi G. (2000), *Wörterbuch Deutsch – Gadertalisch Vocabolar Todësch – Ladin (Val Badia)*, San Martin de Tor, Istitut Ladin “Micurà de Rü”.

Building a Lexicon for a Kernewek MT System

Paul R. Bowden

School of Computing and Mathematics

The Nottingham Trent University

Nottingham, England

NG1 4BU

paul.bowden@ntu.ac.uk

Résumé – Abstract

On présente une description de la création d'un lexique cornouaillais (Kernewek kemmyn). Le lexique comprend des entrées distinctes des formes sujet à la mutation et conjugées. Il est aussi possible d'ajouter les mots nouveaux au lexique. Cette fonction montre les mutations possibles et des affixes connus. Le lexique est utilisé dans un système TM par traduire du cornouaillais à l'anglais. Des développements à l'avenir dans ce système << TM directe >> feront une forte impression sur les informations dans le lexique. Le lexique sera utile aux académiciens et aux pédagoques.

The approach taken to building from scratch a lexicon for Cornish (Kernewek kemmyn) is described. The lexicon will have separate entries for mutated and inflected forms, but has user assistance in the form of an interactive function to add new words to the lexicon. The interactive function gives advice on possible mutations and on known affixes. The lexicon is being used in a new MT system for Cornish to English and future developments in this direct-MT system may impact upon the data stored in the lexicon in various ways. It is hoped that the lexicon will be useful to the wider academic and pedagogical community when it has reached a useful size.

Keywords – Mots Clés

Lexique, Cornouailles, Kernewek, Anglais, mutation, traduction << directe >> par machine.

Lexicon, Cornish, Kernewek, English, mutation, direct-MT, user-interaction.

1 Introduction

Kernewek (Cornish) is a revived P-Celtic (“brythonic”) language undergoing a renaissance in the South West of England, in the area formerly known as Kernow (the current English county of Cornwall). One strongly-supported paradigm for the revived language is Kernewek Kemmyn, *Common Cornish*. This version was largely introduced by George, who has produced printed dictionaries for *kemmyn* (George, 1998). A detailed grammar also exists (Brown, 1984) as well as much teaching material (Page, 1996),

(Brown, 1996). However, although recently work has started on producing tagsets for Cornish - the LER-BIML project at the University of Lancaster; see (Mills, 2002) - computational resources are scarce, especially for *kemmyn*. The author of this paper has recently started work on a Kernewek to English machine translation (MT) system, written in the 'C' programming language, and therefore requires certain computational resources, including a comprehensive lexicon available under the LINUX/UNIX operating systems. This paper discusses the approach taken to building the lexicon, known as *gerva.txt* in the MT system (since *gerva* is Cornish for vocabulary).

2 Pertinent Features of Cornish

Kernewek, like the historically related Welsh and Breton, features *mutation*. This property alters the initial letters of words according to the preceding words, under certain circumstances. This process is not the same thing as inflection (e.g. of word endings) since it does not depend upon the grammatical function of the mutated word. Instead, it probably originally arose to assist in pronunciation. There are five states of mutation in Kernewek, usually known as *original (dictionary)*, *soft (lenition)*, *breathed (spirant)*, *hard (provection)*, and *mixed (normal and after 'th')*. Mutations occur in a variety of circumstances, such as after the definite article (*an*) for certain nouns (mostly feminine singular nouns and masculine plural nouns of persons), or after prepositions, personal pronouns, certain nouns, certain numbers etc. The mutation system is complex and contains many rules and exceptions to those rules, and is given in Table A1 (in the Appendix). Here are a few examples of mutation:

mamm <i>a mother</i>	an vamm <i>the mother</i>
dydh <i>a day</i>	an jydh <i>the day</i>
diw <i>two(f.)</i>	an dhiw <i>the two(f.)</i>
gwra <i>do/make(3,s)</i>	a wra <i>do/make(3,s)</i>
gwari <i>to play</i>	ow kwari <i>playing</i>
teg <i>pretty</i>	benyn deg <i>a pretty woman</i>
gwelav <i>see(1,s)</i>	omma y hwelav <i>here I see</i>
Pennsans <i>Penzance</i>	dhe Bennsans <i>to Penzance</i>
bydhons <i>will be(3,pl)</i>	ple fydhons i ? <i>where will they be ?</i>

The implications of mutation for the lexicon are obvious: it needs to hold, or to be able to allow generation of, all possible mutated forms. Thus a decision must be made regarding which approach to take. This is discussed shortly.

Kernewek also inflects certain words. As in most languages, plural nouns differ from their singular forms. In English the rules are fairly simple, although there are many exceptions (Bowden et al., 1996) such as foot/feet, mouse/mice, die/dice, series/series, tomato/tomatoes etc. Cornish is also quite simple in this respect, with a few exceptions and some central vowel changes, and the usual crop of outright exceptions. The main mechanism for forming a plural noun is by the addition of *ow* (sometimes *yow*) e.g. chi/chiow (house/houses), res/resyow (row/rows of things). Vowel changes may occur

e.g. karr/kerri (car/cars), edhen/ydhyn (bird/birds). Other forms exist e.g. tiek/tiogyon (farmer/farmers) and the use of -s or -ys e.g. plen/plenys (plain/plains). The decision must be made whether to hold separate plural forms in the lexicon, or to generate them from the singular forms as headwords. In addition, Kernewek has a system of collective nouns which can be used to create words for the singular objects which make up the collective.

Although there are two genders in Cornish (*m.* and *f.*, as in French), adjectives do not have to agree with nouns in gender (or indeed in number), and mostly follow their nouns. There is no alteration of word endings with grammatical function (Subject, Object etc) as prepositions and sentential word order are used to signify these aspects (as is done in English, although Cornish uses several standard sentential word orders including both SVO and OVS). Morphologically, the only other major feature of Cornish is the system of verb inflection. There are several indicative tenses in Kernewek, and also a subjunctive mood, each tense having standard inflections dependent upon person and number. These endings are standard (although there are, of course, exceptions and irregular verbs). A regular verb such as *prena* (to buy) may have 50 or so forms sharing the stem (*pren-*). However, since mutation can occur, each of these 50 forms may mutate e.g. *prenas* to *brenas*. For some verbs, such as the auxiliary verb *galloes* (to be able), there is more than one mutation that can occur (g disappears, g to k, and g to h in the case of *galloes*) so that very many orthographic forms may occur in Cornish text. Since the MT system must be able to translate all possible forms, a strategy for doing this needs to be devised.

There is also a system in Cornish for attaching prepositions to personal pronouns, which may be applied to several prepositions in a similar manner. Thus the preposition *rag* (for) may give rise to *ragov* (for me), *ragos* (for you(s)), *ragdho* (for him/it) etc. However, this is a relatively small closed class of words, and so it was decided at the outset to enter each of the preposition+pronoun words as separate lexicon entries. There are probably only tens or possibly a couple of hundred such forms, since not all prepositions support them.

3 Processing and Lexicon Decisions

3.1 Introduction to Decisions Made

In this section we shall discuss the approaches taken to the problems described above. However, it is worth considering the motivation behind the Kernewek MT system at this point, as it bears upon the decisions made. The MT system is partially motivated by the desire to see how far a very shallow approach, based on the ‘direct MT’ paradigm, can be pushed. The direct approach is described in (Hutchins and Somers, 1992) and represents a shallow MT method which starts with lexical transfer and continues with any necessary post-processing stages (e.g. to alter word order in the target language, equivalent to structural transfer.)

Currently, the initial word-for-word transfer is coded, with a small amount of post-processing attempted (much more will be developed in the future). Post-processing currently in place includes *ow* + infinitive to give *-ing* verb form, e.g. *ow donsya* gives rise to “my/(-ing_verb_follows) to_dance” which is processed to “dancing” under certain circumstances. However, this initial stage of development requires first and foremost the provision of a good lexicon. Later stages of processing may well depend upon decisions made regarding the format of lexicon entries, so the lexicon is paramount at this early stage of development. (Having said that, the reverse may also be true – post-processing may necessitate certain lexicon conventions not obvious at present.)

Due to the non-availability of machine-readable Kernewek kemmyn dictionaries, the lexicon is being built by hand, based upon input texts throwing up unknown Cornish words. Thus whenever a new text is input, new words will usually be added to the lexicon as a result. As the system matures, it is hoped that the unknown word rate will stabilise at a low figure per thousand words of input. The mechanism for handling unknown words is described below, as it assists the user with a degree of automation and with supporting information.

3.2 Mutations

It has been decided to include all mutated forms as separate lexicon entries. This will be done initially manually. The impact of this decision may not be known until many tens of thousands of words have been added to the vocabulary. Effects may include reduced processing speeds. If this becomes a problem, reorganisation of the *gerva.txt* file into several smaller files (e.g. indexed by initial letter) may be necessary. Ultimately, this approach *may* have to be abandoned and replaced by a mutation-processor function which is called when needed. Note that the rules regarding possible mutations (as given in Table A1) have already been built into the mutation-assistance code of the unknown-word processor (see Section 4 below). Thus it would not prove too difficult to extend this code into a full de-mutation function in the future. However, this might necessitate the inclusion of other information in the lexicon, which is not currently present. This other information might include the genders of nouns, and the inflection state (2, 3, 4, 5 or 6) produced by prepositions etc on the word that follows.

3.3 Nouns

Nouns of all numbers will have separate entries. The impact here is much smaller than that caused by mutations, although both singular and plural nouns will have separate entries for mutated forms where this can occur.

3.4 Verbs

In a previous MT system, the *Brutus* Latin to English system (Bowden, 2001), the author developed a full morphological processor for verbs (and indeed for all the other inflected parts of speech). This was possible because Latin dictionaries contain enough information attached to each headword verb to allow the generation of the many tens of forms that

each may occur in, and because Latin is generally a very regular highly-inflected language. In Cornish, however, although it is known how most verbs conjugate, it is not usual to indicate this information explicitly in dictionaries. Instead, grammar books may be consulted for general categories of verbs and the inevitable exception lists e.g. (Brown, 2001).

The situation regarding how to handle verb endings is currently as follows. Unless it becomes obvious as the *gerva.txt* file grows that the approach will not work for practical (processing) reasons, e.g. speed of processing when many tens or hundreds of thousands of entries exist in *gerva.txt*, no morphological processor will be developed. As with all other entries, each individual form will have its own entry. Different senses will be separated by the slash character ‘/’ as is standard in the vocabulary file. In addition, number and person information will be attached, with tense indicated by the English meaning. (The irregular verb *bos* (to be) is handled slightly differently, with personal pronouns also attached.)

3.5 Adverbs

In Cornish, adverbs are formed from adjectives using the particle *yn*, or by using the adjective unchanged. The approach taken is not to list adverbs in the lexicon (unless they are common single-word adverbials) but to create them as context dictates in a post-processing stage (e.g. by changing *yn fen* = ‘in strong’ to ‘strongly’). Thus the English ‘strongly’ will not appear anywhere in the lexicon, because this contains only single Cornish words (not pairs of words or larger phrases).

This is similar to the approach taken for *-ing* verb forms, where only the infinitive form of the verb is stored (e.g. *prenas==to_buy* - see above). It remains to be seen how easy it is to correctly recognise and hence process such bigrams in the raw English translation of a Kernewek sentence. There is no obvious reason why this should not be possible, however.

3.6 Hyphenated and Apostrophe Forms

Many orthographic words (strings of characters surrounded by white space) in Cornish contain a hyphen or an apostrophe. All such forms will be listed separately in the lexicon. These include standard contractions (e.g. *p'eur* = *py eur*, *y'n* = *yn an*) and infixed pronouns attached to verbal particles (e.g. *a'th* = *vbl_ptl_a+you(s)*, *na's* = *vbl_ptl_na+her|it*, *y'n* = *vbl_ptl_y+him|it* etc). The former are standard forms and effectively constitute a closed class, and the latter too because the infixed pronouns always attach to one of a small set of verbal particles. Thus it makes sense to list these all as separate lexicon entries. Where one form has more than one sense (e.g. *y'n*) then the slash separator ‘/’ will be utilised to separate them within a single lexicon entry, as is usual.

Since the method of building the lexicon is based upon real texts (i.e. from Kernewek newspapers, books and websites) it is hoped that the most commonly used words will be collected early on, and that rarely-used forms will therefore not clog up the lexicon.

3.7 Examples from The Lexicon

Some examples for all of the above categories are now given, in the form of extracts (non-contiguous) from the current *gerva.txt* file:

```

amari====cupboard
a'm====(vbl_ptl_a)+me/of_me
andhiblansneth====indefiniteness
anedha====from_them
anedhi====from_her|it
a====(Neg_quest?)/goes/if/of/from/(vbl_ptl_a)
a'n====in_the/(vbl_ptl_a)+him|it
anodho====from_him|it
an====the
awelek====very_windy
a-woeless==below
a-wosa====after
awos====because_of
a'y====of_his|her|its
bal====mine/spade
balores====chough
balyow====mines/spades
bara====bread
bargen-tir====farm
bedha====he|she_it_used_to_be/used_to_have(1,s)|(1,pl)|(2,pl)
bedhen====I_used_to_be/let_us_be
beu====he|she|it_was/had(1,s)|(1,pl)|(2,pl)
veu====he|she|it_was
bluvenn====pen
blydhen====year
blydhynyow====years
bodhek====voluntary/volunteer
boken====or_else/either
bons====they_were/they_may_be
bo====or
bos====to_be
bosva====existence
bowessyn====rested(1,pl)
bowgh====you(pl)_may_be
bownder====lane
brav====well_done!
brehg====arm
brehhow====arms
brena====to_buy
brenas====bought(3,s)
brenis====bought(1,s)
brensyst====bought(2,s)
bretonek====Breton
brithel====mackerel
broder====brother
bryntin====splendid
bub====each/every
chekkennow====cheques
chi====house
chiow====houses
chons====luck/chance
daffar====equipment
da====good

```

4 Supporting the Building of the Lexicon

In order to assist the user of the MT system, an interactive mode of running has been provided. When selected, this highlights any unknown Cornish words and requests that the user supply the meaning, in standard *gerva.txt* format. Furthermore, to aid this process, the interactive mode presents to the user a list of “near miss” words that are already in the lexicon. This near-miss system works entirely on spelling differences, and is not semantic in nature. It is currently set up so as to bring up words which are greater than 70% similar to the unknown word in terms of their character string. The algorithm used is the Ratcliff-Obershelp algorithm, which has been coded into a C-function named *how_similar()* which returns a percentage similarity between 0% and 100%. The algorithm is word-length independent (Computing, 1992). The advantage of this approach is that word-endings (e.g. for singular/plural nouns, verb tense/person endings, adjectival endings e.g. *-ek*) are effectively highlighted. Furthermore, mutation (i.e. word-beginning change) is also handled well. The example output given below shows how both of these aspects are usefully handled.

```
SENTENCE 1 TRANSLATING SENTENCE:
Moy es eth kans person a dhe Borth Pyran rag gweles gwari blydhenek .
UNKNOWN WORD DETECTED - PLEASE GIVE MEANING IN GERVA.TXT STANDARD FORM:
SIMILAR WORDS TO Borth IN GERVA.TXT ARE:
'orth' which means 'against/at/(-ing_verb_follows)' (88 percent sim.)
'orthis' which means 'at_you(s)' (72 percent sim.)
'orthiv' which means 'at_me' (72 percent sim.)
'orthyn' which means 'at_us' (72 percent sim.)
'porth' which means 'port' (80 percent sim.)
DE-MUTATION? The word 'borth' MAY be a mutated form, or it may not...
DE-MUTATION: This word starts with 'b', so an unmutated form MIGHT start with 'p'...
(User entered "port" at this point)
OK - will add 'borth==port' to gerva.txt ...
UNKNOWN WORD DETECTED - PLEASE GIVE MEANING IN GERVA.TXT STANDARD FORM:
SIMILAR WORDS TO blydhenek IN GERVA.TXT ARE:
'blydhen' which means 'year' (87 percent sim.)
'vlydhen' which means 'year' (75 percent sim.)
DE-MUTATION? The word 'blydhenek' MAY be a mutated form, or it may not...
DE-MUTATION: This word starts with 'b', so an unmutated form MIGHT start with 'p'...
HINT: This word ends -ek, so might be an adjective/adverb ending -y or -ly, based
upon a preceding noun stem.
(User entered "yearly" at this point)
OK - will add 'blydhenek==yearly' to gerva.txt ...
<omitted material here>

SENTENCE 2 TRANSLATING SENTENCE:
Dy Sul 9/3/03 : Herwyth hengov nowydh , eth kans person eth dhe Borth Pyran hedhyw rag
gweles gwari a-dro dhe vywnans Sen Pyran .
UNKNOWN WORD DETECTED - PLEASE GIVE MEANING IN GERVA.TXT STANDARD FORM:
SIMILAR WORDS TO vywnans IN GERVA.TXT ARE:
'nans' which means 'at_this|that_time' (72 percent sim.)
'veynnas' which means 'wanted(3,s)' (76 percent sim.)
'veynnens' which means 'was_wanting(3,pl)/might_want(3,pl)' (71 percent sim.)
'veynnes' which means 'to_want/were_wanting(2,s)/might_want(2,s)' (76 percent sim.)
'vennis' which means 'wanted(1,s)' (76 percent sim.)
'veynnons' which means 'want(3,pl)/may_want(3,pl)' (71 percent sim.)
'veynsa' which means 'would_want(3,s)' (76 percent sim.)
'veynnses' which means 'would_want(2,s)' (71 percent sim.)
'veynnsys' which means 'wanted(2,s)' (71 percent sim.)
'venn' which means 'wants(3,s)' (72 percent sim.)
'vewnans' which means 'life' (85 percent sim.)
DE-MUTATION? The word 'vywnans' MAY be a mutated form, or it may not...
DE-MUTATION: This word starts with 'v', so an unmutated form MIGHT start with 'b' or 'm'...
(User entered "life" at this point)
OK - will add 'vywnans==life' to gerva.txt ...
```

In the above, the user (who is assumed to know a basic amount of Cornish grammar) is able to deduce with a high degree of certainty that *borth* is probably a mutated form of *porth* (port), that *vywnans* probably means the same as *bywnans* (life) with a b-to-v mutation, and that *blydhenek* probably means ‘yearly’, as it is clearly *blydhen* or *vlydhen* (‘year’) with the added *-ek* ending.

Note how the user is assisted with **DE-MUTATION** information, which applies if the unknown word is a mutated form (the user must consider the possibilities here, as the system does not make a definite decision as to whether the unknown word is mutated or not, but merely points out the possibilities for a word starting with that letter or letters, as given in Table A1). Note also the **HINT**: given by the program, which has several such suggestions available (e.g. for words starting *kes-*, ending *-weydh* etc) Such hints allow the addition of a new entry to the vocabulary without necessarily having to consult a printed dictionary. This is important because, due to mutation, consulting a traditional Kernewek dictionary can be time consuming, as the mutated forms are not usually separately listed. Note that the user simply has to type in the English meaning(s) for the presented unknown Cornish word, in correct *gerva.txt* format, and then press RETURN, and the system then adds this “on the fly” to *gerva.txt* whilst at the same time using it in the sentence currently being translated.

Furthermore, as the *gerva.txt* file grows, it will actually become more and more useful. The mechanism can only present near-miss words it already knows, so as more words are added, more near-miss candidates will be available. Future tests will also result in adjustment of the ‘miss-level’ (currently 70% similarity or better is reported) to optimise performance, i.e. bring back mutated words and inflected forms without bringing up too many unrelated words due to accidental word trunk similarities. This miss-level may also need to vary dependent on the length of the unknown word (number of letters).

One disadvantage of the interactive method described above is that, if a Kernewek word already has a vocabulary entry, then it is used by the MT system without interaction from the user. This is fine so long as *all* the senses of that word are present in the vocabulary entry, separated by slashes. Thus the user must always check the output translation to be alert to cases where a sense is missing from the dictionary entry. Fortunately, this is fairly easy, as a critical missing sense usually renders the output English obviously wrong where the offending word occurs. For example, both the 2nd person singular imperative and the 3rd person singular present/future indicative of verbs are often the same (the bare stem), and it may be that the imperative form is not listed in the *gerva.txt* entry in all cases. However, context will usually eventually reveal such omissions, and the entry can then be corrected manually.

5 Concluding Remarks

It is hoped that the *gerva.txt* file will eventually be released for use by the academic community interested in NLP for Cornish. It is presently too small (less than 5,000

entries) and too unstable in format for release (e.g. the use of ‘/’ and ‘|’ to divide word senses, standard person/tense markers etc are all working solutions that may alter in time as processing needs are uncovered). For example, it may well be that tense/mood information needs to be added to verb entries, due to some as-yet undiscovered post-processing requirement. Currently, part of speech is not indicated at all, except by implication (e.g. a meaning with (3,s) after it must indicate a verb). It may be that this has to change in future. The shallow philosophy used in the system development may in fact prove to be infeasible in certain areas, necessitating extra information attachment to lexicon entries.

It is also possible that once the lexicon reaches a critical size, some pre-processing i.e. standalone programs may be developed which can auto-generate many new entries (e.g. mutated forms, verb forms). The practicality of this approach will depend upon the regularity of the grammatical rules used by Kernewek and the availability of exact descriptions of such rules e.g. in grammar textbooks. It will be done, however, only if it is *necessary* to do it. If it is rare for the system to find a word it does not know, then there would be little point in using a pre-processor to fill the lexicon with tens of thousands of very rare occurrences.

The *gerva.txt* file may also prove useful to Kernewek language learners. Since it should contain *all* word forms, both headword and inflected, and both original and mutated, and since it will contain *all* possible senses for each entry, it could provide the basis for an easily-used dictionary for Cornish to English. With the creation of suitable software, the lexicon should also provide the source for an English to Cornish dictionary. This too would show all mutated forms, e.g. on looking up “port” the dictionary would give both *porth* and *borth*. It is hoped, then, that this project will have a pedagogical impact for Kernewek kemmyn.

References

- Bowden P. R. (2001), Latin to English Machine Translation – A Direct Approach
Machine Translation Review No. 12
<http://www.bcs.org.uk/siggroup/nalatran/mtreview/mtr-12/5.htm>
- Bowden P. R., Halstead P., Rose T. G. (1996), Dictionaryless English Plural Noun Singularisation using a Corpus-based list of Irregular Forms, Proceedings of *ICAME 17* (Stockholm)
- Brown, W. (2001), *A Grammar of Modern Cornish (3rd Edition)* pub. Kesva An Taves Kernewek
- Brown, W. (1996), *Skeul An Yeth – Stus Dhien A’n Yeth Kernewek (parts 1 – 3)* pub. Kesva An Taves Kernewek
- George, K. (1998), *An Gerlyver Kres* pub. Kesva An Taves Kernewek

Hutchins W. J., Somers H. L. (1992), *An Introduction to Machine Translation* pub.
Academic Press

LER-BIML project: <http://www.ling.lancs.ac.uk/biml/bimls3lang.htm>

Mills, J. (2002), Suggestions for a Morphosyntactic Tagset for Cornish, based on the
EAGLES Obligatory and Recommended Attributes
http://www.ling.lancs.ac.uk/biml/cornish_tags.htm

Page, J. (1996), *Grammar for the First Grade/Grammar Beyond the First Grade* pub.
Kesva An Taves Kernewek

Ratcliff-Obershelp Algorithm (1992), Notebook Section, *Computing* (20th August 1992
edition, page 24)

Appendix – Kernewek Mutation Table

1 original (dictionary)	2 lenition (soft)	3 spirant (breathed)	4 provection (hard)	5 mixed (normal)	6 mixed (after ‘th)
B	V		P	F	V
Ch	J				
D	Dh		T	T	T
G + a	-		K	H	H
G + e					
G + i					
G + y					
G + l	-		K		
G + r					
Gw	W		Kw	Hw	W
G + o	Q		K	Hw	W
G + u					
G + ro					
G + ru					
K	G	H			
M	V			F	V
P	B	F			
T	D	Th			

Table A1: Kernewek Kemmyn mutation states (dash indicates dropped initial letter)

Term Extraction for Ladin: An Example-based Approach

Oliver Streiter, Daniel Zielinski, Isabella Ties and Leonhard Voltmer

European Academy, 39100 Bolzano/Bozen, Italy

{ostreiter;dzielinski;ities;lvoltmer}@eurac.edu

Mots-clefs – Keywords

langues minoritaires, extraction terminologique basé sur les exemples
minority languages, example-based term extraction, n-gram similarity

Résumé - Abstract

Cette communication traite le problème de l'extraction de termes pour les langues minoritaires. Nous présentons une méthode basée sur des exemples qui fonctionne même si les ressources linguistiques digitales sont rares. Notre méthode se base sur modèles de termes générés à partir d'un nombre limité de termes d'exemple. Les résultats obtenus pour le Ladin du Val Gherdëna sont meilleurs que ceux des approches statistiques simples à l'extraction de termes.

This paper tackles the problem of Term Extraction (TE) for Minority languages. We show that TE can be realized, even if computerized language resources are sparse. We propose an example-based approach, which draws the knowledge of how a term is formed from a relatively small set of example terms. For the Ladin of Val Gardena, which we use in our experiments, the example-based approach outperforms simple statistical approaches to TE.

1 Introduction

Minority Languages have few speakers, few native linguists and even fewer computational linguists. If those languages have a writing system, they often may not have strict writing rules. They always lack adequate corpora and financial support. Under such conditions, which are the approaches to be followed? Statistical approaches of corpus linguistics need large amounts of language resources. Rule-based approaches (for tagging and parsing) require expensive skilled workers. Technology transfer from other languages often fails if designed specifically for one language or language family. This is the case for shallow NLP techniques which make implicit assumptions about the language. Therefore, example-based approaches seem to be promising. Required are a relatively small number of specific examples which can be created by any native speaker.

Among the most basic needs of a minority language figures the creation and management of terminology. This is the situation for the different idioms of Ladin spoken and written in the Dolomites (Italy). In 1989 Ladin has received official status in the Ladin valleys of Badia and Gardena and in 1993 in Val di Fassa. Since then, legal documents have been written in Ladin.

Term extraction helps creating terminology from texts. We will see now what this task implies, what solutions the scientific literature proposes and how our example-based approach rates amongst them: In Section 2 we describe modern approaches to TE by reference to the unithood-problem and the termhood-problem. In Section 2.3 we describe evaluation techniques for TE. In Section 3 we review the past experiments in the field and propose in Section 3.1 an example-based approach to TE and related this approach to those cited in the literature (Section 3.2).¹

2 Term extraction

2.1 Definitions

TE is an operation which takes as input a document and produces as output a list of term candidates ($\{TC\}$). Term candidates are words or phrases which are potential terms of the subject area represented by the input document. Traditionally, TE is seen as intersection of two problems. The *Unithood Problem* is the task to select language units from a set of word combination (e.g. *red car* but not *is very*). The *Termhood Problem* on the other hand describes the task to select from a set of word combination those combinations which fulfill the requirements of a term (e.g. *red pepper* but not *red car*). More often than not, the unithood problem is solved first and the output TCs are checked for their term-status (the termhood-problem).

2.2 Approaches

Approaches to TE can be classified according to the knowledge used as linguistic or statistic approaches. They may be combined in hybrid approaches in order to join the strong aspects of

¹Abbreviations used in the paper: TC == Term candidate; TCF == Frequency of TC in a given document; DF == Document Frequency, the number of documents with TC; IDF == Inverted Document Frequency = $\frac{1}{DF}$; $\{TC\}$ == Set of term candidates; $\{T\}$ == Term collection == unordered set of terms; $\{T\}_{doc}$ == Subset of T belonging to one specific document; $TC \in \{T\}$ == TC is a term; $\{TC\} \cap \{T\}_{doc}$ == set of correct TCs; $\#\{\dots\}$ == The cardinality of a set.

Table 1: Approaches to TE, an overview.

Linguistic		methods	publications
	intrinsic	POS-tagging,chunking stop-words	(Bourigault & Jacquemin, 1999) (Merkel & Mikael, 2000)
extrinsic	syntagmatic paradigmatic	full parsing term variation	(Arppe, 1995; Soininen <i>et al.</i> , 1999) (Jacquemin, 1999)
Statistical		methods	publications
	intrinsic	mutual information likelihood ratio	(Church & Hanks, 1989) (Hong <i>et al.</i> , 2001)
extrinsic	syntagmatic paradigmatic to document	nc-value entropy c-value weirdness	(Maynard & Ananiadou, 1999) (Merkel & Mikael, 2000) (Nakagawa, 2001) (Brekke <i>et al.</i> , 1996)

complementary approaches. Another, orthogonal, classification describes approaches to TE as intrinsic relative to the TC (e.g. morphological information) or extrinsic relative to the TC. The extrinsic approach may be syntagmatic (e.g. syntactic, contextual information) or paradigmatic (e.g. relations among TCs and $\{T\}$).

Linguistic approaches make use of morphological, syntactic or semantic information implemented in language-specific programs. Its main aim is to identify language units. For reasons of efficiency and accuracy, assumptions on how terms are formed are weaved into the linguistic analysis. These assumptions may refer to the number of words to be combined, special suffixes or part of speech requirements. Morphological analyzers, part-of-speech taggers and parsers are used for this type of analysis. A list of stop-words, e.g. words that might not occur in a specific position of a TC (beginning, middle, end) may be used in addition to other criteria.

Statistical approaches to TE are based on the detection of one or more lexical units in specialized documents with a frequency-derived value higher than a given threshold. They are useful both for extracting single-word and multi-word units. The assumption is that documents are characterized by the repeated use of certain lexical units or morpho-syntactic constructions. We discuss only the more elementary measures.

(1) Frequency of occurrence: The more frequently a lexical unit appears in a given document the more likely it is that this unit has a special function or meaning. Yet extracting TCs just by frequency would also render frequently appearing combinations of function words as TCs. Even in combination with a filter for certain morpho-syntactic patterns, this approach is not always satisfactory.

A second assumption is that linguistic expressions which characterize a document are frequent within a document but infrequent across different documents. One measure to capture this idea is TF.IDF (Equation 2 in Annex), widely used in information retrieval. It divides the TCF_x , the frequency of occurrence of TC_x in the document, by the document frequency DF_x , i.e. the number of other documents which contain TC_x . The same assumption is expressed in the weirdness-ratio (Equation 3) which uses relative frequencies for TF and IDF (Brekke *et al.*, 1996).

The main problem with frequency-based approaches is that they may work well for one-word

units, but do no scale up for two- or three-word units. If the TE-approach, whatever it may be, is based on a sample of 10.000 words, an extension to two-word units would require a 100.000.000 word sample in order to obtain equally good results. For the treatment of tree-word units, $10.000^3 = 1.000.000.000.000$ words would be required. This problem is known as sparse-data problem and is present in all measures which involve the frequency of the TC as undivided unit, e.g. the joint probability.

The frequency of a TC is one type of association measures. Association measures are used to rate the correlation of word pairs. These measures can be derived from the *contingency table* of the word pair (A,B) (Tab. 6). The contingency table contains the observed frequencies of (A,B), (A,notB), (notA,B) and (notA,notB), marked here as $O_{11} \dots O_{22}$. If the occurrences of (A,B), (A,notB) ... are independent, their expected frequencies are estimated from the product of the marginal sums. These are stored as $E_{11} \dots E_{22}$. Lexical association measures are formulas that relate the observed frequencies O to the expected frequency E under the assumption that A and B are independent. The simple frequency corresponds to O_{11} in this table.

Mutual Information (MI) (Equation 4) measures the association between two units. This measure is used frequently in corpus linguistics, even though it works badly for low-frequency events. The MI can be defined as the probability of the joint occurrence of w_1 and w_2 , divided by the product of the probabilities of the singular occurrences. If two words occur once side by side in a one hundred words corpus, they get a MI of $\sim \log_2(100)$. On the other hand, if they co-occur twice, they get a MI of $\sim \log_2(50)$. This shows that the probability of the joint singular occurrence has been rudely overestimated.

Another problem with frequency-based measures is that they may rank TCs correctly if they contain the same number of words, but rank TCs of more words too low or too high. Actually, many measures are simply not defined for measuring the association between more than two words. If we would be forced to create a MI-measure for 3-word units, this could be defined as, starting from a three-dimensional contingency table as in Equation 5. A singular 3-word expression in a 100 word corpus would consequently receive the MI of $\sim 10.000!$

Other, more appropriate measures are e.g. the χ^2 -measure, the *t-score* and the likelihood ratios: The χ^2 -measure for dependence (Equation 6) doesn't assume normally distributed probabilities. Frequencies should be 5 or higher in order to apply the *chi²*-measures. The *t-score* (Equation 7) and the log-likelihood ratio (Equation 8) are better suited for low-frequency data. The latter however is not defined if w_i or w_j appears only in the pair (w_i, w_j) (Daille, 1994).²

To sum up, all frequency-based techniques assign a numerical value to sets of words to rank TCs and to exclude TCs below a certain threshold. The unit-hood problem is not properly addressed for two reasons. First, the measure is applicable only to n -word sequences with a fixed n , e.g. $n = 2$. Secondly, word associations do not respect phrase boundaries, ie. they may identify parts of a phrase or associations as *look at*, where *at* belongs to the following PP.

Another statistical approach aims at the identification of boundaries of TCs. If the boundaries are defined as the first and last word of a TC and the words preceeding and following them, this approach is suitable for TCs of variable length, without requiring larger corpora for the identification of longer TCs. If boundaries are defined as the entire TC and the words preceeding and following it, the problem of sparse data reappears. The boundary is classically gauged via

²Most word association measures are implemented in the Perl-module *N-gram Statistics Package* which can be freely downloaded from CPAN.

the entropy, but any association measure could be used to locate a boundary there where low associations are found.

2.3 Evaluation

Although much of the usefulness of TE depends on the way how TE programs are integrated into the terminographer's working environment, approaches to TE are frequently evaluated in terms of *recall*, *precision*, *mean* and the *ranked recall*.

The *recall* describes the capacity to identify all terms contained in a document. It is defined as the number of correctly identified TCs divided by the number of terms in the text. With a recall of 80%, 20% of the terms remain undetected.

The *precision* describes the accuracy with which words and phrases are classified as terms. If the terminographer has to discard many TCs, the precision is low. The precision is defined as the number of correctly identified TCs divided by the number of all proposed TCs. With a precision of 80%, 20% of the TCs are not terms.

High values of recall often imply low precision scores and vice versa. Therefore recall and precision are frequently combined into the harmonic *mean*.

TE may produce for a medium-sized text many thousands TCs and it is important to rank them. We use the *ranked recall* as a further evaluation criterion. If we define r_i as the rank of the i th $TC | TC \in \{\{TC\} \cap \{T\}_{doc}\}$ then the ranked recall is defined in Equation 1: In a list of 3 TCs with the second and third $TC \in \{T\}_{doc}$, the ranked recall is $\frac{1+2}{2+3} = 0.6$.

3 TE for Minority Languages

(Brekke *et al.*, 1996) explore the potential of the weirdness ratio for TE for small languages, taking Norwegian as an example. They use a 10.000 word specialist text and a 100.000 word general language corpus. Following the limitations of this measure, only one-word units are extracted and ranked. For languages which almost exclusively use compounding for term formation, this method may be adequate. For analytical languages, this method leaves out too many terms as we demonstrate below. The weirdness-ratio has also been applied in (Ahmad & Davies, 1994), in this case to Welsh, with the specialist text and the general language corpus having each the size of 100.000 words.

(Daille *et al.*, 2000) report on two experiments with Malagasy, an Austronesian language in Africa. In the first experiment, a statistical language-independent TE approach (ANA (Enguehard & Pantera, 1994)) has been tested on a corpus of 25.000 words. The system has a good precision (about 75%) but a low recall: only about 240 TCs have been extracted. In a second experiment a hybrid, linguistic and statistical, approach has been tested which required the prior creation of a dictionary and the training of POS-tagger. This required the creation of a dictionary and the training of POS-tagger, before TE could start. With 819 TCs, the number of the extracted TCs is higher than in the purely statistical approach. Precision rates, however, are not reported. This work documents the difficulties of TE with non-European languages. It gives several hints at possible solutions to the question of how linguistic approaches may be put to work even in difficult circumstances.

Table 2: Simple methods for TE, tested individually. The ranking of TCs is done via TF.

Method	#\{TC\}	recall	precision	mean	ranked recall
no method	19019	1	0.0056	0.011	0.011
punctuation	8023	1	0.0134	0.026	0.0179
f-words	6289	0.946	0.016	0.033	0.030
length	2419	0.9375	0.044	0.084	0.055
pattern	489	0.848	0.202	0.326	0.388

3.1 Example-based TE

Example-based approaches in NLP are characterized by the fact that the training material is of the same type as the system's output. Feeding a computer with parse trees to train parsing, is an example-based approach to parsing. Feeding a computer with examples of classified documents to classify documents is an example-based approach to document classification.

The advantage of example-based approaches over rule-based approaches is that no abstract rules are required. As for the acquisition of the data this means that examples can be extracted automatically or created manually by enumerating positive examples. As for the representation, no complex formalisms are required to express the linguistic knowledge. Exceptions and regular phenomena can be listed side by side. The advantage of example-based approaches over statistical approaches is that the system can start even from few training examples.

Tackling TE with an example-based approach requires only a few example terms, e.g. for English *red pepper*, *information society*. These examples can be traditionally elaborated terms in an existing term-base, thus reflecting the properties of terms. In this case, the termhood and unithood problem may be treated conjointly at the same time. TE with an example set of only nominal phrases will produce nominal phrases and TE with an example set containing also verbal phrases will extract also those. If no terms are available, dictionary entries may suffice. These entries may be reworked manually in order to improve TE.

Some non-example-based filters are used for experimentation. The first filter concerns *function words* (f-words). These are identified automatically and used to exclude TCs with function words as first or last word (Merkel *et al.*, 1994). The 100 words of the background corpus with the highest DF are assumed to be function words. A second filter, called *punctuation* assumes that punctuation marks are not part of a TC.

Example-based filters are (1) affix term-patterns (2) upper-case/lower-case (graphic) term patterns and length filters. These can best be explained with an example. The Ladin term *tofla de comune* is transformed into the pattern **a—de—*e*, by reducing non-function words to their last character. The upper-case/lower-case term pattern creates a *c* for capitalized words, an *l* for lower-case words and an *x* otherwise. The term *tofla de comune* generates the graphic pattern *l—l—l*.

Words are extracted as TCs if they fit to one affix and one graphic pattern, even if coming from different examples. The sequences of words *ciasa de comune*, *cuntlameda de comune* etc would then be recognized as TC.

The length of the example terms can be used to calculate a 'good' length of TCs. We defined this good length as the mean (m) of the length of the example terms ± 3 standard deviations.

Table 3: Combination of simple methods for TE.

Method	#{TC}	recall	precision	mean
pattern	489	0.848	0.202	0.326
pattern + punctuation	489	0.848	0.202	0.326
pattern + length	477	0.839	0.203	0.328
pattern + f-words	390	0.839	0.248	0.386

Terms which are too small or too long are therefore filtered out.

3.2 Experiments

The experiments are conducted using a text of 994 words, written in the Ladin variant that is spoken in Val Gardena. The text describes the by-laws of the community and contains, according to a manual examination, 113 terms.

In Table 7 different types of training material are compared with respect to the effects on the TE quality. The training material can be a list of related or unrelated terms, a list of dictionary entries or a mixture of both. The results show that, if terms are used as examples, we get a high precision and a low recall. Using dictionary entries as examples enhances the recall and reduces precision. For the experiments to follow, the mixed method will be used.

Table 2 compares the results of the different TE methods. The first row, 'no method', represents the base line. 19019 TCs are extracted with the perfect recall of 1 and the precision of 0.0056. Assuming that terms never feature punctuation marks, only 8023 TCs are extracted with perfect recall. The following two methods exclude all TCs with function words in first or last position or with extreme length. This filter is not sufficiently specific though, because there are still 20 TCs for one term. The example-based patterns method on the other hand is more selective than any other and retains a good recall (0.85).

Table 3 shows the effect on the results when combining different methods. We start with the example-based method and try to improve its precision. The combination with the function word filter reduces the recall only by 1% but enhances the precision by 4%.

Table 4 gives the results for the weirdness-ratio method, which only extracts single-word terms ($n = 1$). With a recall of 54%, 46% of the terms remain undetected. This might still be a good method for compounding languages or for very fundamental terms. Table 5 shows the results of the TE with Mutual Information with $n = 2$. The recall of Mutual Information with only around 10% is quite low. The results clearly show that free-length approaches are to be preferred over those with a fixed n , because a fixed n drastically reduces the recall without necessarily improving the precision: The best precision value, 0.255, is yet not better than the precision with unrestricted n .

Figure 1, shows the learning curve for example-based TE with the examples coming from (a) a term-list, (b) a word list and (c) a mixture of both. The results show a quick rise in recall. While the recall rises continually, the precision drops after a few hundred examples. Apparently, with more training data, no more good term-models are learned, and those term-models which cause noise accumulate. The automatic identification and exclusion of inappropriate term-models is one possible direction for our future research on example-based TE.

Table 4: Extraction of 1-word TCs with the weirdness ratio.

Method	#TC	recall	precision	mean	ranked recall
weirdness ratio	345	0.544	0.188	0.280	0.363
” ” + pattern	312	0.544	0.210	0.303	0.415
” ” + length	316	0.544	0.205	0.298	0.400
” ” + length + pattern	302	0.544	0.215	0.308	0.416
” ” + f-words	281	0.544	0.225	0.318	0.367
” ” + f-words + pattern	250	0.544	0.254	0.346	0.404
” ” + f-words + pattern + length	249	0.544	0.255	0.347	0.404

Table 5: Extraction of 2-word TCs with Mutual Information.

Method	#TC	recall	precision	mean	ranked recall
MI (2 word terms)	807	0.098	0.013	0.024	0.007
MI + pattern	160	0.098	0.063	0.074	0.064
MI + pattern + f-words	69	0.098	0.144	0.110	0.144

4 Conclusions

In this paper we have shown that example-based term extraction offers a feasible approach to TE for minority languages which only needs few or little resources. A few examples, drawn from dictionaries or other terminological data are sufficient to create term-models which cover most terms to be extracted. The input texts can be very short. While other approaches, especially statistical approaches require large texts, we could extract about 100 terms from a relatively small text of only 1.000 words.

The proposed example-based approach replaces an in-depth linguistic analysis of the input document. Due to this shallowness, the approach is prone to errors resulting from surface similarities of terms and non-terms. In the same way as linguistic approaches, the example-based approach can be combined with sophisticated statistical ratings when large corpora are available and with linguistic tools like stemmers.

The TE tools is freely available under <http://dev.eurac.edu:8080/perl/all.tar.gz>. A graphical interface is provided with Bistro <http://dev.eurac.edu:8080>. Currently we exploit within the Project Logos Gaias (<http://logos-gaias.themenplattform.com>) the integration of the example-base TE tools into GYMN@ZILLA (Streiter *et al.*, 2003), for the purpose of classifying, indexing and pedagogic elaboration of documents.

References

- AHMAD K. & DAVIES A. (1994). 'weirdness' in special-language text: Welsh radioactive chemicals texts as an exemplar. *Journal of the International Institute for Terminology Research*, 5(2), 22–52.
- ARPPE A. (1995). Term extraction from unrestricted text. Helsinki. Short Paper presented at the 10th Nordic Conference of Computational Linguistics (NoDaLiDa).

- BOURIGAULT D. & JACQUEMIN C. (1999). Term extraction + term clustering. An integrated platform for computer-aided-terminology. In *Proceedings of EACL*, p. 15–22. Bergen.
- D. BOURIGAULT, C. JACQUEMIN & M.-C. L'HOMME, Eds. (2001). *Recent Advances in Computational Terminology*, Natural Language Processing, John Benjamins, Amsterdam.
- BREKKE M., MYKING J. & AHMAD K. (1996). Terminology management and lesser-used living languages: A critique of the corpus-based approach. In (*Sandrini, 1996*), p. 179–189.
- CHURCH K. W. & HANKS P. (1989). Word association norms, mutual information and lexicography. In *27th Annual Meeting of the ACL*, p. 76–83, Vancouver.
- DAILLE B. (1994). Combined approach for terminology extraction: lexical statistics and linguistic filtering. Université Paris VII.
- DAILLE B., ENGUEHARD C., JACQUIN C., RAHARINIRINA R. L., RALALAOHERIVONY B. S. & LEHMANN C. (2000). Traitement automatique de la terminologie en langue malgache. In K. C. ET AL., Ed., *Ressources et évaluation en ingénierie des langues, Actualités scientifiques- Universités Francophones*, p. 225–242. De Boek and Larcier s.a.
- ENGUEHARD C. & PANTERA L. (1994). Automatic natural acquisition of a terminology. *Journal of Quantitative Linguistics*, 2(1), 27–32.
- HONG M., FISSAHA S. & HALLER J. (2001). Hybrid filtering for extraction of term candidates from German technical texts. In *Proceedings of Terminologie et Intelligence Artificielle, TIA'2001*, Nancy.
- JACQUEMIN C. (1999). Syntagmatic and paradigmatic representation of term variation. In *ACL'99*, p. 341–348.
- MAYNARD D. & ANANIADOU S. (1999). Identifying contextual information for multi-word term extraction. In (*Sandrini, 1999*), p. 212–222.
- MERKEL M. & MIKAEL A. (2000). Knowledge-lite extraction of multi-word units with language filters and entropy thresholds. In *Proceedings of RIAO*, volume 1, p. 737–746. Paris: Collège de France.
- MERKEL M., NILSSON B. & AHRENBER L. (1994). A phrase-retrieval system based on recurrence. In *Proceedings of the Second Annual Workshop on Very Large Corpora (WVLC-2)*, p. 43–56, Kyoto.
- NAKAGAWA H. (2001). Experimental evaluation of ranking and selection methods in term extraction. In (*Bourigault et al., 2001*), p. 303–325.
- P. SANDRINI, Ed. (1996). *Proceedings of Terminology and Knowledge Engineering (TKE'96)*, Innsbruck. TermNet.
- P. SANDRINI, Ed. (1999). *Proceedings of Terminology and Knowledge Engineering (TKE'99)*, Vienna. TermNet.
- SOININEN P., VOUTILAINEN A. & TAPANAINEN P. (1999). An experiment in automatic term extraction. In (*Sandrini, 1999*), p. 234–241.
- STREITER O., KNAPP J. & VOLTMER L. (2003). Gymn@zilla: A browser-like repository for open learning resources. In *ED-Media, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Honolulu, Hawaii.

Table 6: Contingency table of observed frequencies $O_{11} \dots O_{22}$ for the word pair (A,B) (top) and for estimated frequencies $E_{11} \dots E_{22}$ under independency assumption (bottom).

	$w_2 = B$	$w_2 \neq B$	\sum
$w_1 = A$	O_{11}	O_{12}	R_1
$w_1 \neq A$	O_{21}	O_{22}	R_2
\sum	C_1	C_2	N
$w_1 = A$	$E_{11} = \frac{R_1 * C_1}{N}$	$E_{12} = \frac{R_1 * C_2}{N}$	
$w_1 \neq A$	$E_{21} = \frac{R_2 * C_1}{N}$	$E_{22} = \frac{R_2 * C_2}{N}$	

Table 7: Examples drawn from different resources: Termbanks or Dictionaries.

Method	# {TC}	recall	precision	mean
termbank	299	0.7321	0.284	0.41
dictionary	322	0.75	0.269	0.396
mixture	390	0.839	0.248	0.386

Equations:

$$\text{ranked recall} = \frac{\sum_i^n i}{\sum_i^n r_i} \quad (1)$$

$$\text{TF.IDF}_x = \frac{TCF_x}{DF_x} \quad (2)$$

$$\text{weirdness ratio}_x = \frac{\frac{TCF_x}{\#\{TC\}}}{\frac{DF_x}{\sum_{d=1}^m doc_j}} \quad (3)$$

$$MI = \frac{O_{11}}{E_{11}} \quad (4)$$

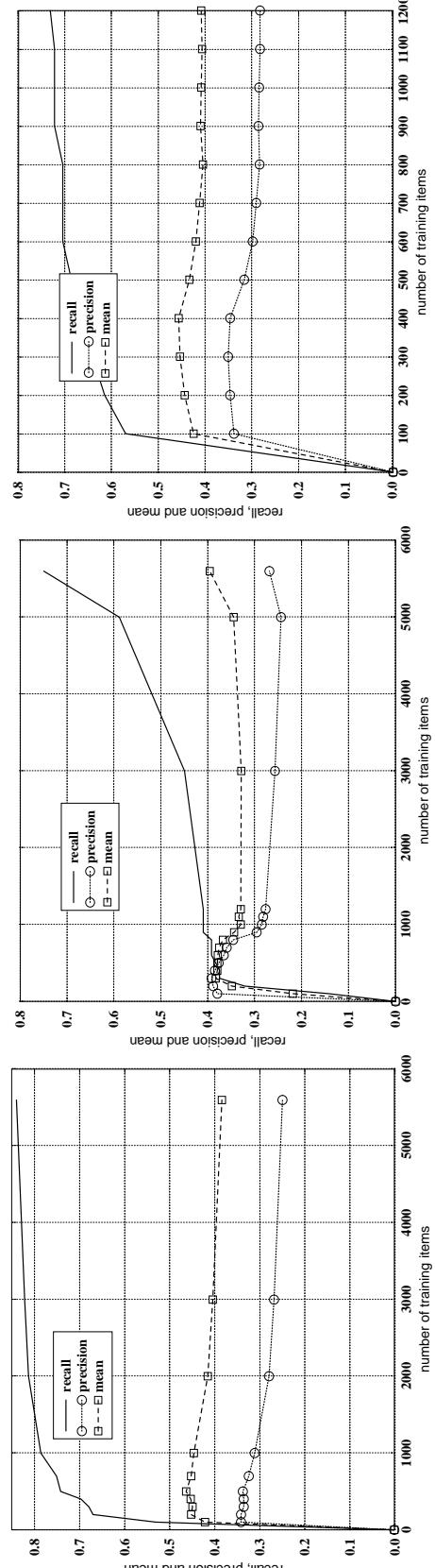
$$MI_3 = \frac{O_{111}}{E_{111}} \quad (5)$$

$$\chi^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (6)$$

$$\text{t-score} = \frac{O_{11} - E_{11}}{\sqrt{O_{11}}} \quad (7)$$

$$\text{log-likelihood} = 2 \sum_{ij} O_{ij} * \log_2 \left(\frac{O_{ij}}{E_{ij}} \right) \quad (8)$$

Figure 1: Learning curves with 1200 example terms (left), 5000 dictionary entries (middle) and a mixture of both (right).



Index des auteurs

Ahmed Abdelali		
Cross-Language Information Retrieval using Ontology	117	
Eneko Agirre		
Disambiguation of case suffixes in Basque	213	
Salah Aït-Mokhtar		
Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques	57	
Atelach Alemu		
Natural Language Processing for Amharic: Overview and Suggestions for a Way Forward	173	
Lars Asker		
Natural Language Processing for Amharic: Overview and Suggestions for a Way Forward	173	
Sophie Aubin		
Evaluation comparative de deux analyseurs produisant des relations syntaxiques	67	
Nathalie Aussenac-Gilles		
Construction d'ontologies à partir de textes	27	
Philippe Blache		
Une grille d'évaluation pour les analyseurs syntaxiques	77	
Evelyn Bortolotti		
Linguistic Resources and Infrastructures for the Automatic Treatment of Ladin Language	253	
Didier Bourigault		
Construction d'ontologies à partir de textes	27	
Paul R. Bowden		
Building a Lexicon for a Kernewek MT System	265	
Marie-France Bruandet		
Mise en place d'un Système de Recherche d'informations en vietnamien	149	
Michael Carl		
Introduction à la traduction guidée par l'exemple (Traduction par analogie)	11	
Jean-Pierre Chevallot		
Mise en place d'un Système de Recherche d'informations en vietnamien	149	
James Cowie		
Cross-Language Information Retrieval using Ontology	117	
Antônio Carlos da Rocha Costa		
SignWriting and SWML: Paving theWay to Sign Language Processing	193	
Ernesto William De Luca		
Example-based NLP for Minority Languages: Tasks, Resources and Tools	233	
A. Diaz de Ilarrazá		
HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities	243	
Graçaliz Pereira Dimuro		
SignWriting and SWML: Paving theWay to Sign Language Processing	193	
David Farwell		
Cross-Language Information Retrieval using Ontology	117	
Sisay Fissaha		
Amharic verb lexicon in the context of Machine Translation	183	
George Foster		
Multilinguisme et question-réponse: adaptation d'un système monolingue	107	

Gil Francopoulo		
TagChunker : mécanisme de construction et évaluation	95	
Caroline Gasperin		
Extracting XML syntactic chunks from Portuguese corpora	223	
V. Gendner		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS	87	
Mesfin Getachew		
Natural Language Processing for Amharic: Overview and Suggestions for a Way Forward	173	
Rodrigo Goulart		
Extracting XML syntactic chunks from Portuguese corpora	223	
A. Gurrutxaga		
HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities	243	
Caroline Hagège		
Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques	57	
Johann Haller		
Amharic verb lexicon in the context of Machine Translation	183	
Stephen Helmreich		
Cross-Language Information Retrieval using Ontology	117	
I. Hernaez		
HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities	243	
Bao-Quoc Ho		
Mise en place d'un Système de Recherche d'informations en vietnamien	149	
G. Illouz		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS	87	
M. Jardino		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS	87	
Olivier Kraif		
Repérage de traduction et commutation interlingue : Intérêt et méthodes	127	
Victor Lascurain		
Disambiguation of case suffixes in Basque	213	
Mikel Lersundi		
Disambiguation of case suffixes in Basque	213	
N. Lopez de Gereñu		
HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities	243	
L. Monceaux		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS	87	
Jean-Yves Morin		
Une grille d'évaluation pour les analyseurs syntaxiques	77	
Thi Minh Huyen Nguyen		
Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens	161	
Bill Ogden		
Cross-Language Information Retrieval using Ontology	117	
P. Paroubek		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS	87	
Luc Plamondon		
Multilinguisme et question-réponse: adaptation d'un système monolingue	107	
Luboš Popelínský		
Disambiguation of case suffixes in Basque	213	

Paulo Quaresma		
Extracting XML syntactic chunks from Portuguese corpora		223
Sabrina Rasom		
Linguistic Resources and Infrastructures for the Automatic Treatment of Ladin Language		253
I. Robba		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS		87
Laurent Romary		
Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens		161
Ágnes Sándor		
Problèmes d'intersubjectivité dans l'évaluation des analyseurs syntaxiques		57
K. Sarasola		
HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities		243
Kevin P. Scannell		
Automatic thesaurus generation for minority languages: an Irish example		203
Oliver Streiter		
Example-based NLP for Minority Languages: Tasks, Resources and Tools		233
Oliver Streiter		
Term Extraction for Ladin: An Example-based Approach		275
Isabella Ties		
Term Extraction for Ladin: An Example-based Approach		275
Jacques Vergne		
Un outil d'extraction terminologique endogène et multilingue		139
Renata Vieira		
Extracting XML syntactic chunks from Portuguese corpora		223
A. Vilnat		
Proposition de protocole d'évaluation des analyseurs syntaxiques du français : PEAS		87
Leonhard Voltmer		
Term Extraction for Ladin: An Example-based Approach		275
Xuan Luong Vu		
Une étude de cas pour l'étiquetage morpho-syntaxique de textes vietnamiens		161
Daniel Zielinski		
Term Extraction for Ladin: An Example-based Approach		275

