COGNITIVE SCIENCE AND KNOWLEDGE MANAGEMENT SERIES

Automatic Text Summarization

Juan-Manuel Torres-Moreno

ISTE

WILEY

Automatic Text Summarization

Series Editor
Jean-Charles Pomerol

Automatic Text Summarization

Juan-Manuel Torres-Moreno

iSTE

WILEY

First published 2014 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2014947781

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-84821-668-6

Contents

Foreword by A. Zamora and R. Salvador

Foreword

The need to identify important information

Throughout history, the sheer amount of printed information and the scarce availability of time to read it have always been two major obstacles in the search for knowledge. Famous novelist W. Somerset Maugham wrote "It was customary for someone who read a book to skip paragraphs, pages or even whole sections of the book. But, to obviate paragraphs or pages without suffering losses was a dangerous and very difficult thing to do unless one had a natural gift for it or a peculiar facility for an on-going recognition of interesting things as well as for bypassing invalid or uninteresting matters."[1] Somerset Maugham called this the art of skipping pages, and even himself had an offer by a North American publisher to re-edit old books in abbreviated form. The publisher wanted him to omit everything except the argument, main ideas and personages created by the author.

[1] *Ten Novels and Their Authors* by W. Somerset Maugham.

The problem of information storage

In the November 1961 issue of the *Library Journal*, Malcolm M. Ferguson, Reference Librarian at the Massachusetts Institute of Technology, wrote that the December 12, 1960, issue of *Time* magazine included a statement that, in his opinion, “may provoke discussion and perplexity”. The article reported that Richard P. Feynman, Professor of Physics, California Institute of Technology (who would later receive a Nobel prize), had predicted that an explosion of information and storage would soon occur on planet Earth and argued that it would be convenient to reduce the amount and size of present information to be able to store all the world’s basic knowledge in the equivalent of a pocket-sized pamphlet. Feynman went on to offer a prize to anyone reducing the information of one page of a book to one twenty-five-thousandth of the linear scale of the original, in a manner that it could be read with an electron microscope.

One year after Ferguson’s article, Hal Drapper from the University of California published a satirical article called “MS FND IN A LBRY” in the December 1961 issue of *The Magazine of Fantasy & Science Fiction*. Drapper poked fun at the idea of trying to cope with Feynman’s predicted information explosion problem by compressing data to microscopic levels to help store the information and by the development of indexes of indexes in order to retrieve it.

The information explosion was and still is a real problem, but the exponential growth in the capacity of new electronic processors overcomes the barrier imposed by old paper archives. Electronic book readers, such as Amazon’s Kindle, can now store hundreds of books in a device the size of a paperback book. Encoding information has even been taken to the molecular level, such as the synthetic organism created through genetic engineering at the J. Craig Venter Institute which used nucleotides in the organism’s DNA to encode a message containing the names of the authors and contributors. And the message would replicate when the organism multiplies.

Automatic size reduction

Since the dawn of the computer age, various attempts have been made to automatically shrink the size of the documents into a human-readable format. Drapper suggested one experimental method which consisted of reducing the cumbersome alphabet to mainly consonantal elements (thus: thr cmbrsm alfbt ws rdsd t mnl cnsntl elmnts) but this was done to facilitate quick reading, and only incidentally would cut down the mass of documents and books to solve the information explosion. More sophisticated methods attempted to identify, select and extract important information through statistical analysis by correlating words from the title to passages in the text, and by analyzing the position in which sentences occurred in the document trying to assign importance to sentences by their positions within the text. We (Antonio Zamora and Ricardo Salvador) worked at Chemical Abstract Service (CAS), where abstracting and indexing performed manually was our daily job[2]. Realizing that it was difficult to recognize what was important in a document, we developed a computer program that started by trying to discover what was not important, such as clichés, empty phases, repetitive expressions, tables and grammatical subterfuges that were not essential for understanding the article. Our technique to eliminate non-significant, unessential, unsubstantial, trivial, useless, duplicated and obvious sentences from the whole text reduced the articles to the salient and interesting points of the document.

By the late 1970s, we could manufacture indicative abstracts for a fraction of a dollar. These abstracts contained 60–70% of the same sentences chosen by professional abstractors. Some professional abstractors began to worry that they could lose their jobs to a machine. However, primary journals started providing abstracts prepared by the authors themselves; so there was no demand for automatic abstracting. The Internet has changed all that. Many news feeds with unabridged text have become available that can overwhelm anyone looking for

[2] See sections 1.5 and 3.1.4.

information. Yes, presently there is a real need for automatic abstracting.

The future

Today’s smart phones have more computational power than many mainframe computers of the 20th Century. Speech recognition and automatic translation have evolved from being experimental curiosities to tools that we use every day from Google. We are at the threshold of artificial intelligence. IBM’s Watson program won a one-million dollar Jeopardy contest against the two best human champions. Cloud-based computing has removed all the constraints of memory size from our portable devices. It is now possible to access extensive knowledge bases with simple protocols. The ease with which dictionaries can be accessed allows us to use synonyms in our contextual searches so that “bird flu” will retrieve “avian influenza”. Great advances in automatic abstracting have been made during the last 40 years. It is quite possible that in the next quarter of a century there will be computer programs with enough cognition to answer questions such as “What were the important points of this article?”. This is exactly what automatic abstracting strives to accomplish. For specific tasks, the behavior of these new programs will be indistinguishable from that of humans. These expectations are not only dreams about a distant future ... we may actually live to see them become reality.

This book by Juan-Manuel Torres-Moreno presents the approaches that have been used in the past for automatic text summarization and describes the new algorithms and techniques of state-of-the-art programs.

Antonio ZAMORA
Ricardo SALVADOR
August 2014

Foreword by H. Saggion

Automatic Text Summarization

Juan-Manual Torres-Moreno

Text summarization, the reduction of a text to its essential content, is a task that requires linguistic competence, world knowledge, and intelligence. Automatic text summarization, the production of summaries by computers is therefore a very difficult task. One may wonder whether machines would ever be able to produce summaries which are indistinguishable from human summaries, a kind of Turing test and a motivation to advance the state of the art in natural language processing. Text summarization algorithms have many times ignored the cognitive processes and the knowledge that go into text understanding and which are essential to properly summarize.

In *Automatic Text Summarization*, Juan-Manuel Torres- Moreno offers a comprehensive overview of methods and techniques used in automatic text summarization research from the first attempts to the most recent trends in the field (e.g. opinion and tweet summarization).

Torres-Moreno makes an excellent job of covering various summarization problems, starting from the motivations behind this interesting subject, he takes the reader on a long research journey that spans over 50 years. The book is organized into more or less traditional topics: single and multi-document summarization, domain-specific summarization, multi-lingual and cross-lingual summarization. Systems, algorithms and methods are explained in detail often with illustrations, assessment of their performances, and limitations. Torres-Moreno pays particular attention to intrinsic summarization evaluation metrics that are based on vocabulary comparison and to international evaluation programs in text summarization such as the Document Understanding Conference and the Text Analysis Conference. Part of the book is dedicated to text abstracting, the ultimate goal of text summarization research, consisting of the production of text summaries which are not a mere copy of sentences and words from the input text.

While various books exist on this subject, Torres-Moreno's covers interesting system and research ideas rarely cited in the literature. This book is a very valuable source of information offering in my view the most complete account of automatic text summarization research up to date. Because of its detailed content, clarity of exposition, and inquisitive style, this work will become a very valuable resource for teachers and researchers alike. I hope the readers will learn from this book and enjoy it as much as I have.

Horacio SAGGION
Research Professor at Universitat Pompeu Fabra
Barcelona, August 2014

Notation

The main notations used in this book are the following:

V	the set of words in a text
$\|V\| = N$	the cardinality of the set V (N words)
$s \in D$	an element s belonging to the set D
$A \backslash B$	the complement of sets A and B elements of A without the elements of B
λ	a linear combination parameter, $0 \leq \lambda \leq 1$
P	the precision of a system
R	the recall of a system
$\\|\boldsymbol{x}\\|$	the norm of the vector $\boldsymbol{x}$: $\\|\boldsymbol{x}\\| = \sum_i x_i^2$
$C(\bullet)$	the occurrence number of an event
C_w^s	the occurrences of the word w in the sentence s
$\mathcal{C}$	independent and identically distributed (i.i.d.) set of observations: $\mathcal{C} = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$
$\mathcal{LL}(\mathcal{C})$	the log-probability of observations: $\mathcal{LL}(\mathcal{C}) = \log(\mathcal{L}(\mathcal{C}))$
ρ	the number of sentences of a document
$\vec{D} = (s_1, s_2, \ldots, s_\rho)$	a document of ρ sentences (VSM model)
$\vec{Q}$	a query of N terms, $Q = (q_1, q_2, \ldots, q_N)$
Sum	a summary of a document

Π	a set of n summaries $\Pi = \{\text{Sum}_1, \text{Sum}_2, \ldots, \text{Sum}_n\}$
ε	an empty word
w	a word over an alphabet Σ
$\|w\|$	the length of the word w ($\|\varepsilon\| = 0$)
s	a sentence composed of N words
s_μ	a sentence $\mu \subset \{1 \ldots \rho\}$
$\vec{s_\mu}$	a sentence vector $\mu \subset \{1 \ldots \rho\}$ (VSM model)
$\|s\| = \|w_1, w_2, \ldots, w_N\|$	the length of a sentence of N words w_j
w_j	a word of s in positions $j \subset \{1 \ldots \|s\|\}$
$\omega(s)$	the normalized weight of the sentence s, $0 \leq \omega(s) \leq 1$
$\omega_{i,j}$	the weight of the word j in the sentence i
σ	a discourse segment $\in s$
$p_e(\sigma)$	the probability of eliminating a discourse segment σ
$adj[s]$	adjacency matrix of the sentence s (VSM model)
$S[i,j]$	sentences × terms matrix (VSM model)
$\text{sim}(x,y)$	function that quantifies the similarity between 2 objects
τ	compression rate (characters, words or sentences)
ϵ	error of tolerance for convergence
$\mathcal{D}_{JS\|KL}(A\|\|B)$	Jensen–Shannon (JS) and Kullback–Leibler (KL) divergence between two probability distributions A and B

Introduction

Tormented by the cursed ambition always to put a
whole book in a page, a whole page in a sentence,
and this sentence in a word. I am speaking of myself[1]
Joseph Joubert (1754–1824), *Pensées, essais et maximes.*
Gallica http://www.bnf.fr

The need to summarize texts

Textual information in the form of digital documents quickly accumulates to huge amounts of data. Most of this large volume of documents is unstructured: it is unrestricted text and has not been organized into traditional databases. Processing documents is therefore a perfunctory task, mostly due to the lack of standards. Consequently, it has become extremely difficult to implement automatic text analysis tasks. Automatic text summarization (ATS), by condensing the text while maintaining relevant information, can help to process this ever-increasing, difficult to handle, mass of information.

Summaries are the most obvious way of reducing the length of a document. In books, abstracts and tables of content are different ways

[1] *"S'il est un homme tourmenté par la maudite ambition de mettre tout un livre dans une page, toute une page dans une phrase, et cette phrase dans un mot, c'est moi"*.

of representing the condensed form of the document. But what exactly is a text summary? The literature provides several definitions. One definition states that the summary of a document is a reduced, though precise, representation of the text which seeks to render the exact idea of its contents. Its principal objective is to give information about and provide privileged access to the source documents. Summarization is automatic when it is generated by software or an algorithm. ATS is *a process of compression with loss of information*, unlike conventional text compression methods and software, such as those of the gzip family[2]. Information which has been discarded during the summarization process is not considered representative or relevant. In fact, determining the relevance of information included in documents is one of the major challenges of automatic summarization.

The summarization process

For human beings, summarizing documents to generate an adequate abstract is a cognitive process which requires that the text be understood. However, in a few weeks interval, the same person could write very different summaries. However, after an interval of several weeks, the same person can write very different summaries. This demonstrates, in part, the difficulty of automating the task. Generating a summary requires considerable cognitive effort from the summarizer (either a human being or an artificial system): different fragments of a text must be selected, reformulated and assembled according to their relevance. The coherence of the information included in the summary must also be taken into account. In any case, there is a general consensus that the process of summarizing documents is, for humans, a difficult cognitive task.

Fortunately, automatic summarization is an application requiring an extremely limited understanding of the text. Therefore, current systems of ATS have set out to replicate the results of the abstracting process and not the process itself, of which we still have a limited understanding.

[2] For more information, see http://www.gzip.org/.

Although great progress has been made in automatic summarization in recent years, there is still a great number of things to achieve.

From the user's perspective, people are not always looking for the same type of summary. There is also another type of user: automatic systems which use the results of a summarization system as the foundation for other tasks. Many different types and sources of documents exist (both textual and/or multimedia), such as legal, literary, scientific and technical documents, e-mails, tweets, videos, audio and images. As a result, there is no such thing as one type of summary. Sources and user expectations have prompted many applications to be created. Even for text documents, there is a large number of automatic summarization applications in existence (for people or machines):

– generic summarization;

– multi-document summarization;

– specialized document summarization: biomedical, legal texts, etc.;

– web page summarization;

– meeting, report, etc., summarization;

– biographical extracts;

– e-mail and e-mail thread summarization;

– news, rich site summary (RSS) and blog summarization;

– automatic extraction of titles;

– tweets summarization;

– opinion summarization;

– improving the performance of information retrieval systems, and so on.

Automatic text summarization

ATS became a discipline in 1958 following H.P. Luhn's research into scientific text summarization. Two or three important works [EDM 61, EDM 69, RUS 71] were completed before 1978, but they were followed by some 20 years of silence. In the early 1990s, however, the works of K. Spärck-Jones and J. Kupieck improved this landscape. Currently, ATS is the subject of intensive research in several fields, including natural language processing (NLP) and other related areas.

ATS has benefited from the expertise of a range of fields of research: information retrieval and information extraction, natural language generation, discourse studies, machine learning and technical studies used by professional summarizers. Answers have been found to several questions concerning ATS, but many more remain unsolved. Indeed, it appears that 50 years will not suffice to resolve all the issues concerning ATS. For instance, although generating a summary is a difficult task in itself, evaluating the quality of the summary is another matter altogether. How can we objectively determine that the summary of one text is better than another? Does a "perfect" summary exist for each document? What objective criteria should exist to evaluate the content and form of summaries? The community is yet to find answers to these questions.

About this book

Since 1971, roughly 10 books have been published about document summarization: half of these are concerned with automatic summarization. This book is aimed at people who are interested in automatic summarization algorithms: researchers, undergraduate and postgraduate students in NLP, PhD students, engineers, linguists, computer scientists, mathematicians and specialists in the digital humanities. Far from being exhaustive, this book aims to provide an introduction to ATS. It will therefore offer an overview of ATS theories and techniques; the readers will be able to increase their knowledge on the subject.

The book is divided into two parts, consisting of four chapters each.

– ☞ I) Foundations:
  - Chapter 1. Why Smmarize Texts?
  - Chapter 2. Automatic Text Summarization
  - Chapter 3. Single-Document Summarization
  - Chapter 4. Guided Multi-Document Summarization
– ☞ II) Emerging Systems:
  - Chapter 5. Multi- and Cross-Lingual Summarization
  - Chapter 6. Source and Domain-Specific Summarization
  - Chapter 7. Text Abstracting
  - Chapter 8. Evaluating Document Summaries

The conclusion and two appendices complete this book. The first appendix deals with NLP and information retrieval (IR) techniques, which is useful for an improved understanding of the rest of the book: text preprocessing, vector model and relevance measures. The second appendix contains several resources for ATS: software, evaluation systems and scientific conferences. A website providing readers with examples, software and resources accompanies this book: http://ats.talne.eu.

This book is first and foremost a pragmatic look at what is eminently an applied science. A coherent overview of the field will be given, though chronology will not always be respected.

Juan-Manuel TORRES-MORENO
Laboratoire Informatique d'Avignon
Université d'Avignon et des Pays de Vaucluse
France, August 2014

PART 1

Foundations

1

Why Summarize Texts?

In the 1780s, Joseph Joubert[1] was already tormented by his ambition to summarize texts and condense sentences. Though he did not know it, he was a visionary of the field of automatic text summarization, which was born some two and a half centuries later with the arrival of the Internet and the subsequent surge in the number of documents. Despite this surge, the number of documents which have been annotated (with Standard Generalized Markup Language (SGML), Extensible Markup Language (XML) or their dialects) remains small compared to unstructured text documents. As a result, this huge volume of documents quickly accumulates to even larger quantities. As a result, text documents are often analyzed in a perfunctory and very superficial way. In addition, different types of documents, such as administrative notes, technical reports, medical documents and legal and scientific texts, etc., have very different writing standards. Automatic text analysis tasks and text mining[2] [BER 04, FEL 07, MIN 02] as exploration, information extraction (IE), categorization and classification, among others, are therefore becoming increasingly difficult to implement [MAN 99b].

1.1. The need for automatic summarization

The expression "too much information kills information" is as relevant today as it has ever been. The fact that the Internet exists in multiple languages does nothing but increase the aforementioned

[1] http://en.wikipedia.org/wiki/Joseph_Joubert

[2] "Text mining, also referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning" (source: Wikipedia: http://en.wikipedia.org/wiki/Textmining).

difficulties regarding document analysis. Automatic text summarization helps us to efficiently process the ever-growing volume of information, which humans are simply incapable of handling. To be efficient, it is essential that the storage of documents is linked to their distribution. In fact, providing summaries alongside source documents is an interesting idea: summaries would become an exclusive way of accessing the content of the source document [MIN 01]. However, unfortunately this is not always possible.

Summaries written by the authors of online documents are not always available: they either do not exist or have been written by somebody else. In fact, summaries can either be written by the document author, professional summarizers[3] or a third party. Minel *et al.* [MIN 01] have questioned why we are not happy with the summaries written by professional summarizers. According to the authors there are a number of reasons: "[...] because the cost of production of a summary by a professional is very high. [...] Finally, the reliability of this kind of summary is very controversial". Knowing how to write documents does not always equate with knowing how to write *correct* summaries. This is even more true when the source document(s) relate to a specialized domain.

Why summarize texts? There are several valid reasons in favor of the – automatic – summarization of documents. Here are just a few [ARC 13]:

1) Summaries reduce reading time.

2) When researching documents, summaries make the selection process easier.

3) Automatic summarization improves the effectiveness of indexing.

4) Automatic summarization algorithms are less biased than human summarizers.

[3] The term "summarizer" will henceforth be used to refer to an agent (either a human or an artificial system) whose role is to condense one or several documents into a summary.

5) Personalized summaries are useful in question-answering systems as they provide personalized information.

6) Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of texts they are able to process.

In addition to the above, the *American National Standards Institute*[4] (ANSI) [ANS 79] states that “a well prepared abstract enables readers to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether they need to read the document in its entirety”. Indeed, in 2002 the SUMMAC report supports this assertion by demonstrating that “summaries as short as 17% of full text length sped up decision-making by almost a factor of 2 with no statistically significant degradation in accuracy” [MAN 02].

1.2. Definitions of text summarization

The literature provides various definitions of text summarization. In 1979, the ANSI provided a concise definition [ANS 79]:

DEFINITION 1.1.– *[An abstract] is an abbreviated, accurate representation of the contents of a document, preferably prepared by its author(s) for publication with it. Such abstracts are useful in access publications and machine-readable databases.*

According to van Dijk [DIJ 80]:

DEFINITION 1.2.– *The primary function of abstracts is to indicate and predict the structure and content of the text.*

According to Cleveland [CLE 83]:

DEFINITION 1.3.– *An abstract summarizes the essential contents of a particular knowledge record, and it is a true surrogate of the document.*

[4] http://www.ansi.org

Nevertheless, it is important to understand that these definitions describe summaries produced by people. Definitions of automatic summarization are considerably less ambitious. For instance, automatic text summarization is defined in the Oxford English dictionary [5] as:

DEFINITION 1.4.– *The creation of a shortened version of a text by a computer program. The product of this procedure still contains the most important points of the original text.*

Automatically generated summaries do not need to be stored in databases (unlike the ANSI summaries) as they are generated online in accordance with users' needs. After all, the main objective of automatic summarization is to provide readers with information and give him or her exclusive access to the source literature [MOE 00, MAN 01]. Nevertheless, summarizing text documents is a process of *compression*, which involves the loss of information. The process is automatic when it is carried out by an algorithm or a computer system. But what information should be included in the summary that is provided to the user? [SPÄ 93, SPÄ 97]. An intuitive answer would be that the generated summary must contain the most important and representative content from the source document. But how can we measure the representativeness and the significance of the information? This is one of the key questions automatic text summarization algorithms are trying to answer.

Karen Spärck-Jones and Tetsuya Sakai [SAK 01] defined the process of generating automatic text summaries (or abstract process) in their 2001 article as follows:

DEFINITION 1.5.– *A summary is a reductive transformation of a source text into a summary text by extraction or generation.*

According to Radev *et al.* [RAD 00]:

DEFINITION 1.6.– *Text Summarization (TS) is the process of identifying salient concepts in text narrative, conceptualizing the*

[5] http://www.oed.com.

relationships that exist among them and generating concise representations of the input text that preserve the gist of its content.

In 2002, Radev *et al.* [RAD 02a] introduced the concept of multidocument summarization and the length of the summary in their definition:

DEFINITION 1.7.– *[A summary is] a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that.*

According to Horacio Saggion and Guy Lapalme [SAG 02b], in terms of function, a summary is:

DEFINITION 1.8.– *A condensed version of a source document having a recognizable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source.*

The ratio between the length of the summary and the length of the source document is calculated by the compression rate τ:

$$\tau = \frac{|\text{Summary}|}{|\text{Source}|} \qquad [1.1]$$

where $| \bullet |$ indicates the length of the document in characters, words or sentences. τ can be expressed as a percentage.

So what is the optimal value for the compression rate? The ANSI recommends that summaries be no longer than 250 words [ANS 79]. [BOR 75] indicate that a rate of $\tau = 10\%$ is desirable for summaries. In contrast, [GOL 99] maintain that the length of the summary is not connected to the length of the source text. Finally, [RAD 02a] and [HOV 05] specify that the length of the summary must be less than half of that of the source document. In fact, [LIN 99]'s study shows that the best performances of automatic summarization systems are found with a compression rate of τ = 15 to 30% of the length of the source document.

We are now going to introduce a definition of automatic summarization, inspired by [HOV 05] [6], which takes the length of the source document into account:

DEFINITION 1.9.– *An automatic summary is a text generated by a software, that is coherent and contains a significant amount of relevant information from the source text. Its compression rate τ is less than a third of the length of the original document.*

Generating summaries demands that the summarizer (both human and algorithm) make an effort to select, reformulate and create a coherent text containing the most informative segments of a document. The notion of segments of information is left purposefully vague. Summaries can be guided by a particular profile, topic or query, as is the case for multidocument summarization [MAN 99a, MOR 99, MAN 01. Finally, coherence, cohesion as well as the temporality of the information presented must also be respected.

Many different types of document summarizations exist. There are two main reasons for this: first, there are many different types and sources of documents and, second, people have different uses for summaries and are, thus, not looking for the same type of document summarization. In any case, there is a general consensus that the process of summarizing a document requires a person to engage significant cognitive effort. To understand this process, it is interesting to take a look at how people produce summaries. What are the precise steps professional summarizers follow in their work? In his book, "The Art of Abstracting", Edward Cremmins [CRE 96] shows us that professional summarizers take 8 to 12 minutes to summarize a standard scientific article. Clearly, it requires considerably less time to summarize a text than to understand it. Other studies have backed up Cremmins' findings. [MIN 01] indicates that a professional summarizer – when he or she is a specialist in the field – can

[6] *A summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s) and that is no longer than half of the original text(s).*

summarize a text of 10 or so pages in 10 or so minutes, though when the text is outside his or her area of expertise, it takes him or her around an hour to complete. In his studies about professional human summarizers [CRE 93, CRE 96], Cremmins identifies two phases in the summarization process: *local analysis* (the content of a sentence) and *global analysis* (content which is connected through several sentences).

Often both phases of the analysis are linked to the summarizer's prior knowledge of a topic; therefore, it is difficult to create an algorithm for this process [END 95a, END 95b, END 98]. Rath, Resnick and Savage published surprising results about professional summarizers [RAT 61]. Four professional summarizers produced summaries and there was a 25% overlap rate [7] between the summary and the source document; after a six month interval the overlap rate fell to 55% for a summary produced by the same summarizer. What made the summarizer change his or her sentence choice over these six months?

This question cannot be answered directly, but hypotheses about the variability and plasticity of language can be put forward. In Poibeau's words [POI 11]: "there is a number of ways of saying the same thing, or, at least, slightly different things, which will nevertheless be interpreted in an almost identical fashion [...] I can [...] express the same idea in two different ways after an interval of just one minute". This underlying plasticity is what makes languages so rich, yet it is simultaneously the reason they are so difficult to process automatically.

In simple terms, the human production of summaries involves two phases: understanding the source text and writing a concise and shortened version of it. Both understanding the text and the production of summaries require linguistic and extra-linguistic skills and knowledge on the part of the summarizer. An approximation of this process is illustrated in Figure 1.1.

[7] The number of identical sentences between two documents.

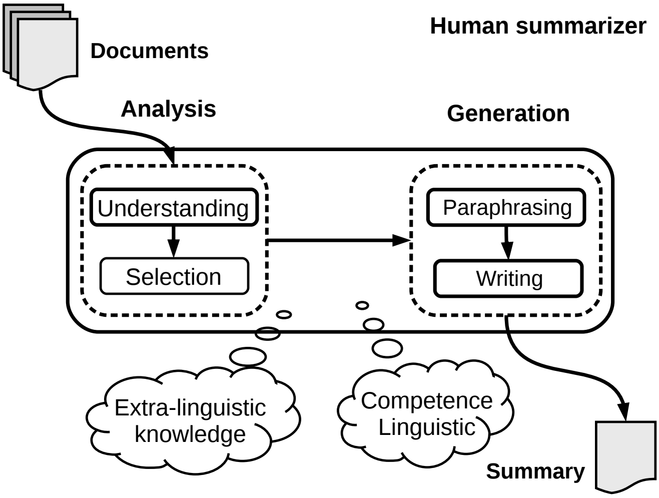


Figure 1.1. *Humans' production of summaries*

Is it worth modeling and trying to replicate the abstracting process of humans? Probably not. The mechanisms of this highly complex process are not very well understood and, what is more, the level of technology it requires is not currently available. Fortunately, other studies suggest that the results of the abstracting process can be approximated by algorithms. What is even more interesting is that automatic summarization (and the overall quality of content), though a long way off summaries produced by people, already do have exploitable properties. Better still, some algorithms can quickly and efficiently process large volumes of documents. Although people produce better summaries (in terms of readability, content, form and conciseness), automatic summarization is accessible, can be quickly distributed and costs very little. Automatic text summarization is an interesting alternative – rather than a replacement – to manual summarization [GEN 09b].

1.3. Categorizing automatic summaries

Summaries have two main functions: direct functions and indirect functions. The direct functions provide an overview of the document (essential information), update the user (update summary), eliminate language barriers (multi- and cross-lingual summary) or facilitate information retrieval (IR). The indirect functions enable documents to be classified and indexed, keywords to be extracted and so on.

Summaries can be categorized according to different sets of criteria, such as their function, type and document source, etc. [MOE 00, MAN 01]:

– According to their function [VAS 63, EDM 69, HOV 05]:

- *indicative summary*: an indicative summary provides information about the topics discussed in the source document. It resembles a table of contents;

- *informative summary*: an informative summary aims to reflect the content of the source text, possibly explaining the arguments. It is a shorted version of the document.

– According to the number of documents for summarization:

- single document: a summary of one document;

- multidocument: a summary of a not necessarily heterogeneous group of documents often about a specific topic.

– According to genre of document:

- *news summary*: a summary of news articles;

- *specialized*: a summary of documents relating to a specialized domain (science, technology, law, etc.);

- *literary*; a summary of narrative documents, literary texts, etc;

- *encyclopedic*: a summary of encyclopedic documents, such as Wikipedia;

- *social networks*: a summary of blogs and very short documents (such as tweets).

– According to type:

- *extract*: an assembly of fragments from the original document;

- *abstract*: summarizing by reformulating: a summary produced by rewriting and/or paraphrasing the text;

- *sentence compression*: a summary containing the same number of sentences as the original text, but with a reduced sentence length.

– According to the type of summarizer:

- *author summary*: a summary written by the author of the document which reflects his or her point of view;

- *expert summary:* a summary written by somebody other than the author, somebody who specializes in the domain but probably does not specialize in producing summaries;

- *professional summary*: a summary written by a professional summarizer, who probably does not specialize in the field, but who has mastered the techniques of writing, norms and standards of producing summaries.

– According to context:

- *generic summary*: a summary of a document which ignores users' information needs;

- *query-guided summary*: a summary guided by information needs or by users' queries [8];

- *update summary*: when users are familiar with a particular topic it is presumed that they have already read documents and their summaries relating to this topic. Therefore, update summaries only show important new information and avoid repeating information.

– According to the target audience:

- *without a profile*: a summary which is independent of the needs and profile of the user; the summary is based uniquely on information from the source documents;

- *based on a user profile*: summaries targeted at users interested in a specialized domain (chemistry, international politics, sports, the economy, etc.).

State-of-the-art systems are able to generate indicative and informative summaries. Informative summaries are considerably harder to produce than indicative summaries as they require that the information contained in the source text be properly understood, generalized, organized and synthesized. However, they can be

[8] *A Generic summary provides the author's point of view, while a Query-based summary focuses on material of interest to the user: http://www.cs.cmu.edu/ref/mlim/chapter3.html.*

substituted at the source, unlike indicative summaries which merely enable an idea of the content of the texts to be produced.

[GEL 99] defined another type of summary: topic selection summary. This type of summary is supposed to generate a succinct report centered around the topics – rather than the ideas – discussed in a document. For instance, one document may talk about science, technology and transport whereas another may discuss politics, war and so on.

Currently, the scientific community is conducting research into the production of multidocument (mono- or multilingual) summaries guided by a query, update summaries and summaries of specialized domains. International campaigns run by the *National Institute of Standards and Technology* (NIST)[9], *Document Understanding Conferences* (DUC) held from 2001 to 2007 and *Text Analysis Conferences* (TAC) held from 2008, have done a great deal to build interest in automatic summarization tasks, by implementing a rather formal framework of testing, creating corpora according to tasks and evaluating participating systems.

Figure 1.2 shows a basic approximation of the process of generating automatic text summaries. The source text (single or multidocument) is processed by a summarization system according to a query and certain parameters, such as the compression rate, the desired type of summary and the desired function. The system produces an extract, an abstract or a summary via compression. The summarization module in the center of the figure will be discussed in more detail in the following chapter.

1.4. Applications of automatic text summarization

There are countless applications of automatic text summarization. Here are just a few:

– increasing the performance of traditional IR and IE systems (a summarization system coupled with Question-Answering (QA) system);

[9] http://www.nist.gov.

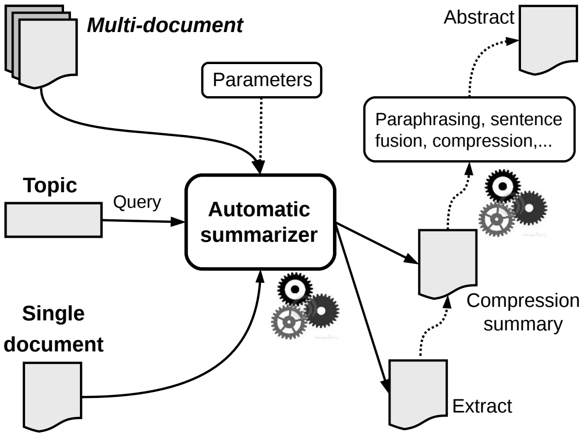


Figure 1.2. *Simplified abstracting process. The source documents and the topic go in. The parameters are the compression rate τ, the type, etc. Extracts, abstracts or sentence compression summaries come out*

– news summarization and Newswire generation [MCK 95a, MAN 00];

– Rich Site Summary (RSS) feed summarization [10];

– blog summarization [HU 07];

– tweet summarization [CHA 11, LIU 12];

– web page summarization [BER 00, BUY 02, SUN 05];

– email and email thread summarization [MUR 01, TZO 01, RAM 04];

– report summarization for business men, politicians, researchers, etc.;

– meeting summarization;

– biographical extracts [MCK 01, DAU 02, HAR 03];

[10] "RSS (Rich Site Summary); originally RDF Site Summary; often dubbed Really Simple Syndication, uses a family of standard web feed formats to publish frequently updated information: blog entries, news headlines, audio and video. An RSS document (called "feed", "web feed" or "channel") includes full or summarized text, and metadata, like publishing date and author's name" (source: Wikipedia: http://en.wikipedia.org/wiki/RSS).

– automatic extraction and generation of titles [BAN 00];

– domain-specific summarization (domains of medicine, chemistry, law, etc.) [FAR 05, BOU 08c];

– opinion summarization [11] [LIU 09], etc.

In addition to these applications, the scientific challenge of replicating a difficult cognitive task, namely the human abstracting process, must also be mentioned. In fact, research aims to make a machine understand a text written in natural language. This would enable, at least theoretically, real – relevant, correct, coherent and fluid – abstracts of source documents to be produced.

1.5. About automatic text summarization

The discipline of automatic text summarization originated with Hans Peter Luhn's research in the 1950s. Already worried about the ever-growing volume of information available at the time, Luhn built a summarizer for scientific and technical documents in the field of chemistry. After this pioneering research, several founding works were written including Edmundson and Wyllys' research in 1961 and 1969 [EDM 61, EDM 69], Rush *et al.'s* work in 1971–1975 [RUS 71, POL 75] and Gerald Francis DeJong's studies in 1982 [DEJ 82]. In 1993, research took off once again with work by Spärck-Jones [12] [SPÄ 93] and Julian Kupiec *et al.* [KUP 95]. This research helped to respark an interest in automatic summarization.

Though incomplete, Figure 1.3 shows the landmark works, books and conferences in the field of automatic summarization. This work will look at these accomplishments in more detail. Below, in italics, is a list of outstanding works in the domain of automatic text summarization:

– "Summarizing Information" by Endres-Niggemeyer [END 98];

[11] See TAC 2008 *Opinion Summarization Task*: http://www.nist.gov/tac/2008/summarization/op.summ.08.guidelines.html.

[12] Karen Spärck-Jones (born August 26, 1935 in Huddersfield, Yorkshire, died April 4, 2007) was a British scientist specializing in NLP. She was one of the pioneers of IR.

– “Advances in Automatic Text Summarization” by Mani and Mayburi [MAN 99a];

– “Automatic Indexing and Abstracting of Document Texts” by Moens [MOE 00];

– “Automatic Summarization (Natural Language Processing)” by Mani [MAN 01];

– “Automatic Summarization” by Nenkova and McKeown [NEN 11].

TIPSTER and SUMMAC’s campaigns in 1997 and 1998 [MAN 02] as well as the NII Testbeds and Community for Information access Research (NTCIR) workshop held in Japan in 2001 and 2002 enabled automatically produced summaries to be evaluated on a large scale (see section 8.4). In 2005, the *Multilingual Summarization Evaluation* (MSE)[13] conducted an international evaluation of Arabic and English summaries. NIST conferences, DUC (2001–2007), TAC from 2008 and CLEF-INEX from 2011 have taken over this role (see section 8.6).

Automatic text summarization is currently the subject of intensive research, particularly, though not exclusively, in Natural Language Processing (NLP). In fact, automatic summarization has benefited from the expertise of several related fields of research (see Figure 1.4) [BRA 95, IPM 95, MAY 95, MCK 95b, SPÄ 95]. Among these fields of research, computer science (understood in a broad and transversal sense), artificial intelligence, and more specifically its symbolic and cognitive methods, have helped summarization [DEJ 79, DEJ 82, ALT 90, BAR 99]. Cognitive sciences have studied the abstracting process [FAY 85, LEM 05]. [RAU 89, RAD 03, RAD 04, TOR 02], IE [PAI 93, RIL 93, MIT 97, RAD 98], Natural Language Generation (NLG) [MCK 93, MCK 95a, JIN 98] and Machine Learning (ML) [ARE 94, KUP 95, LIN 95, LIN 97, TEU 97, MAN 98] approaches have provided automatic summarization with several models. Discourse analysis studies, such as Rhetorical Structure Theory (RST) [MAN 88], have built linguistic models for text summarization, [MAR 00b, CUN 08]. Finally, [CRE 93, END 95a, END 95b,

[13] http://en.wikipedia.org/wiki/DARPA_TIDES_program.

CRE 96, END 98]'s studies have clarified the techniques employed by professional summarizers when generating summaries.

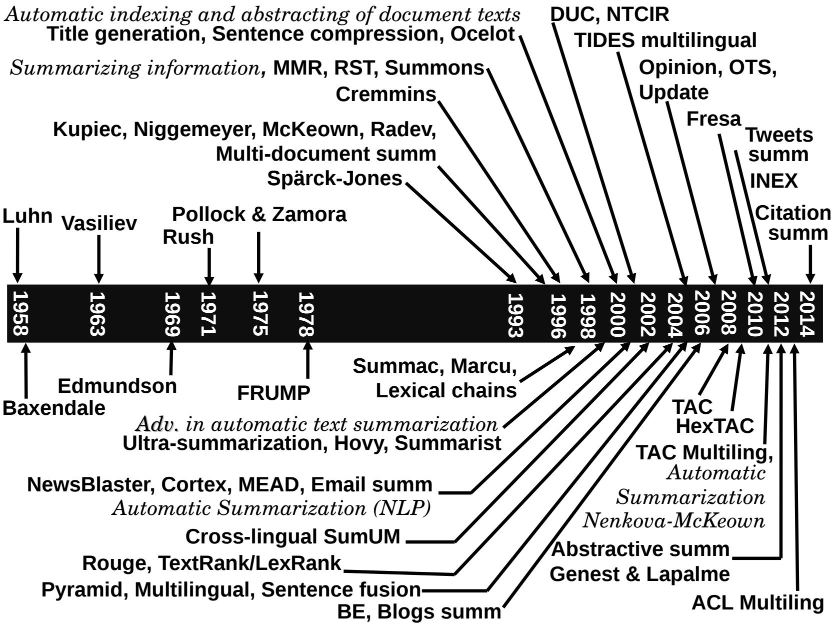


Figure 1.3. *Highlights of automatic text summarization*

One final important point must also be considered with regard to automatic summarization. Knowing how to construct efficient systems which produce summaries is not enough: it is also necessary to know how to evaluate the *quality* of the summaries generated. Therefore, coming up with a clear definition of quality in relation to automatic summarization is essential. In IR (such as web search engines), the performance of a system is evaluated by the relatively objective measures of precision, recall and F-measure. This enables the relevance ranking of documents retrieved by a system in relation to users' queries to be quantified.

However, in automatic text generation, sentence compression and automatic text summarization, what goes in and comes out of the system are documents written in natural language. Therefore, it is difficult to evaluate the relevance of the results. In the concrete case of text summarization, how can we know if the summary produced by an algorithm is correct? How can we determine whether one algorithm produces higher quality summaries than another? Worse still, what does *correct* mean in this applied area of science? What is the ratio of quality and the length, content and form of summaries? In recent years, the scientific community has attempted to develop ways of evaluating quality, but to date no definitive answers have been found. This topic will be picked up in Chapter 8.

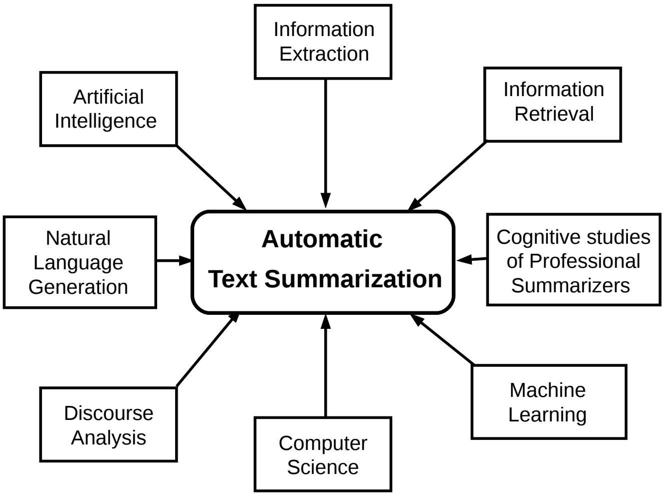


Figure 1.4. *Fields of research which have influenced the development of automatic text summarization*

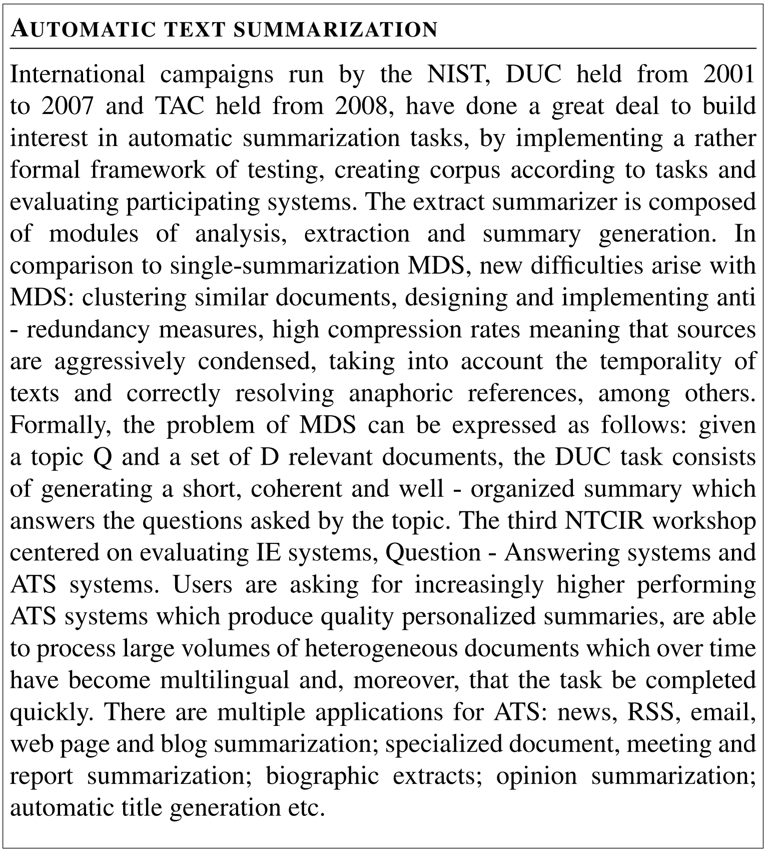
In a little homage to J. Joubert, who was tormented by the ambition to condense a whole book into a page, we have created two summaries of this work. The first (see Figure 1.5) is an author summary of 154 words (compression rate of approximately 0.12 %). The second (Figure 1.6) summary has been generated by a system from (`untex`-ted) LaTeX sources of this book. The text as a whole (apart from the tables, figures and appendices) were automatically produced by the CORTEX algorithm (see section 3.7). The summary was generated in less than 6 s and did not undergo any manual processing. [14]

> **AUTOMATIC TEXT SUMMARIZATION**
>
> Automatic text summarization is a discipline of Natural Language Processing (NLP) which aims to condense text documents. Condensing the text signifies producing a shortened version of the source document which contains the main points of this document. The process involves the loss of information. The author will provide an overview of approaches adopted over time, from Luhn in 1958 to present. Several methods which have been proposed to resolve the issues of automatic summarization will be studied: single and multi-document summarization algorithms, cross- and multi-lingual summarization, specialized summarization, tweets and email summarization, automatic sentence compression and rhetorical analysis summarization. The open-ended and difficult issue of summary evaluation will also be discussed, making reference to both manual and automatic approaches. Several applications of automatic text summarization will also be shown. This work is aimed at people interested in the techniques and algorithms of NLP (students, researchers, linguists, computer scientists, mathematicians, engineers etc.).

Figure 1.5. *Author summary of this work (8 sentences, 154 words, compression rate* $\tau = 0.17$ %*)*

According to the GNU/Linux `wc` command, the source document contained 3,982 sentences and 71,143 words. The compression rate was fixed at seven sentences (307 words), representing a rate of approximately 0.27%.

[14] Conducted on a machine with GNU/Linux v13.10, CPU Intel Core^{TM} i5-3317U$\times$4, 1.70GHz and 4 Gb of RAM.



AUTOMATIC TEXT SUMMARIZATION

International campaigns run by the NIST, DUC held from 2001 to 2007 and TAC held from 2008, have done a great deal to build interest in automatic summarization tasks, by implementing a rather formal framework of testing, creating corpus according to tasks and evaluating participating systems. The extract summarizer is composed of modules of analysis, extraction and summary generation. In comparison to single-summarization MDS, new difficulties arise with MDS: clustering similar documents, designing and implementing anti - redundancy measures, high compression rates meaning that sources are aggressively condensed, taking into account the temporality of texts and correctly resolving anaphoric references, among others. Formally, the problem of MDS can be expressed as follows: given a topic Q and a set of D relevant documents, the DUC task consists of generating a short, coherent and well - organized summary which answers the questions asked by the topic. The third NTCIR workshop centered on evaluating IE systems, Question - Answering systems and ATS systems. Users are asking for increasingly higher performing ATS systems which produce quality personalized summaries, are able to process large volumes of heterogeneous documents which over time have become multilingual and, moreover, that the task be completed quickly. There are multiple applications for ATS: news, RSS, email, web page and blog summarization; specialized document, meeting and report summarization; biographic extracts; opinion summarization; automatic title generation etc.

Figure 1.6. *Automatic summary of this work generated by the* CORTEX *system (7 sentences, 224 words, compression rate* $\tau = 0.17$ %*)*

An automatic evaluation of the quality of the summaries [15] gives the artificial summary a score of 0.00965 and the author summary a score of 0.00593. The reader, of course, has the final say and can judge for himself or herself the relevance of the summaries produced.

[15] See FRESA automatic evaluation in section 8.9.2. The FRESA score falls between $[0, 1]$: 0 indicates a bad summary and 1 indicates that the summary is very similar to the source text.

1.6. Conclusion

Until we have found a solution to the difficult task of making a machine automatically understand text, automatic summarization will remain a – poor – approximation of human summarization. That said, the automatic text summarization methods available have come a long way and achieved significant goals. It is likely to take at least another 50 years to resolve all the issues concerning text summarization. So we can ask ourselves the following questions. How sophisticated should automatic summarization systems be? Is understanding fundamental to the generation of "real" summaries created by reformulating sentences (and not merely providing an extract)? Can statistical methods accomplish this task or is something else required? What should the role of linguistic tools and resources be in automatic text summarization systems? How can we break out of the extraction stage and start to generate real abstracts? How can we objectively measure the overall quality and sensitivity of the content of summaries and the automatic systems that produce them?

This work aims to show how the scientific community has attempted to tackle and find answers to these difficult questions. The answers found are both theoretical – formulating models – and practical – developing technological artifacts and real automatic text summarization systems which are applied to concrete tasks.

2

Automatic Text Summarization: Some Important Concepts

The aim of an automatic Text Summarization system is to generate a condensed and relevant representation of the source documents. The representation, which is in fact a compression involving the loss of information, must preserve the main points of the original document. However, there are countless and heterogeneous sources of information that are capable of being summarized: videos, images, sound and text documents. This work will focus on text document summarization. A summary can be produced from one or several documents. When producing a generic summary, each topic is given the same level of importance during processing; when generating a guided summary, only the information desired by the user is processed. Finally, summarization tasks can select and analyze new information from a set of temporized events and even be multilingual. This chapter will provide an overview of automatic text summarization: preprocessing, types of summary, their uses and generation algorithms. An introduction to the problems related to the evaluation of summaries will also be provided.

2.1. Processes before the process

Automatic text summarization is a complex process that must be broken down into modules if it is ever to be mastered. One such module is text preprocessing. Preprocessing enables a sequence of bits, i.e. a plain text document, to be transformed into an object with minimal linguistic features, such as words and sentences. Preprocessing has two objectives: word normalization and lexicon reduction in a text.

Both of these functions are extremely important for the vectorization of documents (section 2.1.1), because the space of representation is considerably reduced and becomes easier to manage. Therefore, preprocessing is essential, as it provides summarization systems with a clean and adequate representation of the source documents.

In simplified terms, (almost obligatory) preprocessing often involves the following stages:

– splitting the text into segments, sentences, paragraphs, etc.;

– splitting segments into words or tokenization [1];

– word normalization (lemmatization, stemming etc.) [2];

– filtering stopwords, etc [3].

Optional preprocessing can include:

– annotation via grammatical tagging or parts-of-speech (POS)-tagging;

– named entity recognition (NE) [4];

– extraction of terms and keywords;

– weighting terms (with the binary weight, number of occurrences or others according to the representation, see section 2.1.1 and appendix A.2.1), etc.

Figure 2.1 shows a diagram of standard preprocessing that can be applied to source documents before the summarization process.

[1] http://en.wiktionary.org/wiki/token.

[2] Lemmatization and stemming are two processes enabling the *stem* of words to be found. See Appendix A.1.2 for more information.

[3] "In computing, stop words are words which are filtered out prior to, or after, processing of natural language data (text). There is not one definite list of stop words which all tools use and such a filter is not always used. Some tools specifically avoid removing them to support phrase search", (source: Wikipedia: http://en.wikipedia.org/wiki/Stopwords).

[4] "Named-entity recognition (NER) (also known as entity identification and entity extraction) is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc." (source: Wikipedia: http://en.wikipedia.org/wiki/Named-entity_recognition).

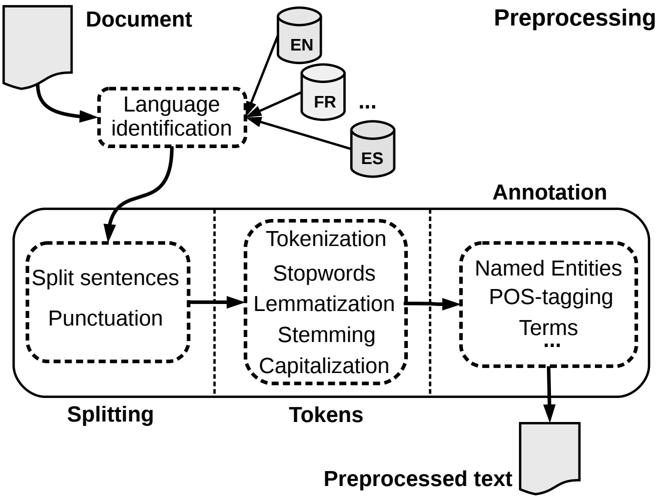


Figure 2.1. *Standard text preprocessing*

Preprocessing is a difficult task that depends to a large extent on the language in which the text is written. Sentence boundaries, for example, are demarcated by punctuation, context and quotations, the use of which varies considerably from language to language. Moreover, not all languages separate words with spaces. In fact, there are considerable differences between splitting a text written in a European language in Latin characters and an eastern language such as Chinese, Japanese or Korean. This problem is also encountered in other stages such as the normalization of words through lemmatization or stemming, POS tagging[5] or NE recognition (see Appendices A.1.5 and A.1.4). Even eliminating stopwords (articles, conjunctions, prepositions, etc.) is not a simple task.

[5] "In corpus linguistics, part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context – i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph" (source: Wikipedia: http://en.wikipedia.org/wiki/Part-of-speech_tagging).

When the entry documents are written in an unknown language it is useful to use an automatic language identification module[6]. The TREETAGGER tool [SCH 94] has become very popular for POS-tagging: it is available in several languages and is based on machine learning (ML) and decision trees[7]. Often involved in preprocessing, word count, word tokens (occurrences of a word) and word types (unique word as a dictionary entry) are essential for statistical methods, particularly for calculating the weight of words.

Preprocessing is also used in other natural language processing tasks, such as the thematic classification of documents. Automatic summarization systems give poor results when the documents to be summarized have not been correctly preprocessed. More information about preprocessing is available in Appendix 1 and in [PAL 10], which contains an excellent chapter about preprocessing.

2.1.1. *Sentence-term matrix: the vector space model (VSM) model*

Once preprocessing is complete and the summarization method has been chosen, a representation of the documents is required. The VSM [SAL 68, SAL 83, SAL 89] is widely used in information retrieval (IR) to represent the documents and terms in an adequate space (see appendix A.2). Several automatic summarization algorithms use this model. In this presentation, we are going to adapt this IR oriented model, defined with N terms and D documents, to a single document with N terms and ρ sentences. Transposition remains valid.

In this model, word order is not important. It is known as the *bag-of-words model*. Each word (term) is given a weight ω, which measures its importance in the document. There are three types of weighting (see Appendix A.2):

[6] See, for instance, The Comprehensive Perl Archive Network (CPAN)'s language identification algorithms: http://search. cpan.org/ambs/Lingua-Identify-0.56/lib/Lingua/Identify.pm or https://metacpan.org/ pod/Lingua::Identify.

[7] TREETAGGER is available at: http://www.cis.uni-muenchen.de/schmid/tools/TreeTagger/.

Binary: $\omega_{\mu,j}$ equals 1 if the term j is in the sentence μ, and 0 if not;

Frequency: number of occurrences of the term j in a sentence μ (term frequency, tf):

$$\omega_{\mu,j} = \text{tf}_{\mu,j}$$

Corrective: using a correction frequency function to take into account the distribution of the word in sentences (inverse document frequency, idf).

$$\omega_{\mu,j} = \text{tf}_{\mu,j} \times \text{idf}_j; \quad \text{idf}_j = \log_2 \frac{\rho}{D(j)}$$

idf_j measures the importance of the term j in the set of ρ sentences and $D(j)$ is the number of sentences in the document in which the word j appears.

Thus, a matrix $S_{[\rho\times N]}$ of ρ rows (sentences) and N columns (terms in the lexicon) is generated. Vectorization transforms a document into a set of ρ vectors sentence $\vec{s}_\mu = (s_{\mu,1}, s_{\mu,2}, \ldots, s_{\mu,N})$. Each sentence is represented by a vector $\vec{s}_\mu$. A lexicon (reduced through simplification during preprocessing) of N words produces a sentence-term matrix $S = [s_{\mu,j}]\,;\ \mu = 1, \ldots, \rho; j = 1, \ldots, N$:

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\rho,1} & s_{\rho,2} & \cdots & s_{\rho,N} \end{pmatrix}; \; s_{\mu,j} = \begin{cases} \omega_{\mu,j} & \text{when the word } j \in \vec{s}_\mu \\ 0 & \text{otherwise} \end{cases} \qquad [2.1]$$

where each row μ contains the weighting $\omega_{\mu,j}$ of the word j in a sentence $\vec{s}_\mu$. The matrix $S_{[\rho\times N]}$ is a representation of the important information contained in the source document in an exploitable format. Graph-based models, when ρ nodes represent the sentences s_μ and an edge $a_{k,\mu}$ of values of similarity (cosinus, Jaccard, Dice, etc.) between sentences s_k and s_μ, are also used a great deal in automatic summarization (see section 3.5).

2.2. Extraction, abstraction or compression?

[MAN 01] defines a summary as a document containing several text units (words, terms[8], sentences or paragraphs) that are not present in the source document. However, this definition is too restrictive as during sentence extraction it is only possible to apply the selection process to fragments of sentences. The resulting extract will, therefore, be composed of units that are not presented in exactly the same way as they were in the original document [BOU 08a]. An intermediate type of text, the condensed text, exemplifies the process used by the majority of summarization systems: important sentences are identified and extracted, and subsequently assembled and reformulated [MAN 01]. In English, there is no ambiguity between the meaning of *summary*, and the meaning of *abstract* or *extract*:

– a summary is a reduced representation that keeps the essential content of a text;

– an abstract is a summary produced by reformulating sentences;

– an extract is a summary produced by extracting sentences from the source text.

However, in languages such as French or Spanish, misuse of the word *"résumé"* (or *"resumen"* in Spanish) has led to the meanings of summary and abstract being conflated and resulting in ambiguity.

A carefully selected title of a document can be considered as the *maximum* summary of a text. In fact, the task of automatically generating titles is related to automatic summarization. It consists of producing the title of a document or a paragraph from the content of the document or paragraph in question. The title must be short, contain the important information from the source and, most importantly, be concise. However, the task of automatically generating adequate titles is very complex. For more information on the topic, see the works of [GÖK 95, BAN 00, JIN 03].

[8] "Term, a lexical unit, word or expression symbolizing a concept" (source: Wikipedia: http://en.wikipedia.org/wiki/Terminology).

When discussing automatic text summarization we very often talk about the text *corpus*. A *corpus* is a large collection (sample) of documents in written or spoken language. The corpus provides a platform for analyzing language, verifying linguistic theories and conducting Machine Learning (ML) on Natural Language Processing (NLP) algorithms for specific tasks. As this work is only concerned with written documents, we will not be looking at recognition, transcription and speech summarization tasks. Written *corpora* are often annotated with linguistic markers or metainformations. This is useful for automatically learning (ML) the characteristics of the *corpora*. However, the majority of documents online are not in the form of structured or annotated *corpus*, rather they are simple collections or sets of heterogeneous documents. Misuse of language has led to these sets also being known as *corpora*. Relevant information can be filtered from the text *corpus* by generating a condensed and summarized presentation of the content of the sources. Automatic summarization algorithms, applied to one – or several – text *corpus*, can filter and generate a reduced and relevant report of the content of the documents.

There are three families of automatic text summarization algorithms:

– summarization by extraction;

– summarization by abstraction;

– summarization by sentence compression.

The main difference between abstraction and extraction lies in the different philosophies about summarization. According to the philosophy of the “good student”, a deep understanding of the document is a prerequisite to produce a quality summary that preserves the content of the text and the author’s intentions. On the other hand, the “bad student” method merely seeks to identify sentences that seem important (due to their position and the occurrence of words) and then extract them. Sentence compression is a specific type of algorithm that seeks to reduce the length of sentences. This technique results in a sentence compression rate of about 33%, which is comparable to a human compression rate. In postprocessing, the compression process can be combined with an extraction approach. This is an attempt to move toward abstraction.

Abstraction seeks to produce summaries by reformulating and fusing ideas (expressed in sentences) and often by using a new lexicon. In extraction, summaries are generated by extracting several sentences that are deemed to be important, just like the “bad students”. But, can extraction approaches be considered as *“bad students”*? Not necessarily, and although they do have major limitations, they also have very significant advantages. One such advantage is that, though unlike the human process, extraction is easier to implement with software. As will be seen throughout this work, extraction algorithms have turned out to be prolific and very high performing.

2.3. Extraction-based summarization

Extraction consists of selecting units of text (sentences, segments of sentences, paragraphs or passages), deemed to contain a document’s essential information (informative content or *informativity*), and of assembling these units in an adequate way. [JIN 00b, JIN 02] observes that 81.5% of sentences in manually generated summaries are borrowed verbatim from the source text. [COP 04] also observes that 70.0% of sentences in summaries appear, at least in part, in the original text. An extract is the assembly of fragments that have been extracted from a source document. The aim of an extract is to give an overview of the original text’s content. The length of this overview can be determined by the compression rate: in other words, “how much shorter is the summary than the original?” [FLU 99].

According to Radev *et al.* [RAD 02b], algorithms for automatic summarization by extraction can be categorized into three types: surface-level, intermediate-level and deep parsing techniques. It is a simple idea: a document is split into lexical units (sentences) that are weighted with a heuristic; the units with the highest scores are then extracted and assembled to create a summary. Figure 2.2 shows a general and mechanized diagram of the automatic summarization process by extraction. The extract summarizer is composed of modules of analysis (preprocessing the document and sentence weighting), extraction (selecting sentences) and summary generation (postprocessing sentences and final assembly). Of course, this is just an abstraction: real systems often contain several layers of software.

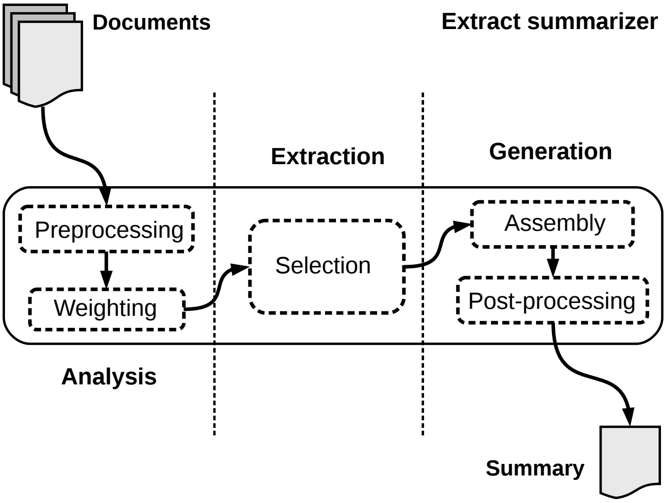


Figure 2.2. *General architecture of an extraction-based summarization system*

2.3.1. *Surface-level algorithms*

Surface-level algorithms do not delve into the linguistic depths of a text; rather they use certain linguistic elements to identify the most relevant segments of a document. Used since the very first studies on summarization, surface-level techniques use the occurrences of words to weight sentences [LUH 58] (see section 3.1.1). Another surface-level technique is based on the idea that words used in the title are important. Extra weight is, therefore, given to sentences containing these words [EDM 69]. Several algorithms use the position of fragments of text. This technique is used for documents with a fixed structure, such as titles, sections, paragraphs and so on. Several studies [BRA 95, LIN 97, HOV 99] have in fact shown that the first lines are always the most important in journalistic articles.

Edmundson 's study [EDM 69], which is concerned with producing indicative extracts, gave weight to index-expression (words or key-sentences). These expressions indicate the importance of a sentence, for instance, “it is important to stress that...” or “by way of conclusion...” (see section 3.1.3). In contrast, other techniques focus on expressions that suggest the unimportance of a sentence, such as “as for example” [TEU 99]: these sentences will consequently not be included in the summary. Other indicators of relevance are key-sentences and certain verbs, which are often used to express important concepts in scientific articles. [PAI 90, POL 75] used this type of marker with good results.

Automatic summarization systems that use statistical techniques to extract information in order to obtain quantitative data about the text also exist. For instance, Berger and Mittal [BER 00] developed OCELOT, a system for synthesizing web pages that uses probabilistic and ML models (see section 6.6.2). Rau *et al.* [RAU 94] developed a system that automatically creates summaries from an important *business news* service, which covers 41 different publications. [DUN 93] proposed the application of log-likelihood $\mathcal{LL}(\mathcal{C})$[9] (likelihood ratio test) with the hypothesis that units in a *corpus* are binominally distributed rather than normally distributed.

There are also systems that essentially use ML. For instance, Kupiec *et al.* [KUP 95] developed an automatic text summarization system based on a Naïve Bayes Classifier[10] that gave good results (see section 3.2.1). In 1998, Inderjeet Mani and Eric Bloedorn developed one of the first algorithms for generic and guided summarization by ML [MAN 98]. Their system was trained on a – small – *corpus* of 188 articles and their abstracts from the Computation and Language E-Print Archive to learn the determining characteristics of sentences preserved

[9] “In statistics, maximum-likelihood estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model’s parameters.” (source: Wikipedia: http://en.wikipedia.org/wiki/Maximum_likelihood).

[10] “A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be *independent feature model*.” (source: Wikipedia: http://en.wikipedia.org/wiki/Naive_Bayes_classifier).

in summaries. The optimal position policy (OPP) [LIN 97] provides empirical validation for the importance of the position of the sentence in a discourse structure. OPP employs sentence weighting using ML. Alonso and Fuentes [ALO 03] developed an extraction summarization system that uses an ML approach followed by sentence compression (sentences are extracted using handwritten rules) to obtain short and grammatically correct sentences.

Finally, other algorithms (that mostly use statistical methods but without learning) are based on the properties of the vectorial model for representing texts (VSM) or on graphs. For example, CORTEX is a summarization system with a decision algorithm that combines several metrics [TOR 02] (see section 3.7). TEXTRANK and LEXRANK [ERK 04, MIH 04b] represent documents as (directed or undirected) graphs in order to identify relevant sentences (vertices) according to their similarity links (see section 3.5.4.2). ENERTEX [FER 07] is a summarizer that makes use of the ideas of statistical physics: it codes documents like a spin system and then calculates the textual energy of sentences as a weighting measure (see section 3.9.3).

2.3.2. *Intermediate-level algorithms*

Intermediate-level algorithms use linguistic information that is more sophisticated than surface-level algorithms but less sophisticated than deep parsing. One intermediate-level technique is lexical chain recognition. Lexical chains are sequences of words connected by lexical semantic relations [BAR 97, SIL 00]. In general, producing an automatic summary involves four stages. First, the original text is divided into segments and then lexical chains are constructed. The "strong" chains are identified, and the sentences containing these strong chains are extracted and assembled to create a summary.

One of the main problems of summarization by extraction is the presence of unresolved anaphoric references [11]. A summary formed of

[11] "In linguistics, anaphora is the use of an expression the interpretation of which depends upon another expression in context (its antecedent or postcedent). In the

a set of extracted sentences can contain anaphoras referring to sentences that have not been extracted by the system. Faced with this problem, several authors [PAI 90, BOG 97, NAN 99, LAL 02] developed techniques to identify and resolve anaphoric coreferences in a bid to improve the quality of extracts. More specifically, [BOG 97] used an automatic resolution system for anaphoric references that calculated a salience measure (in terms of grammatical, syntactic and contextual parameters) for words that had previously been identified as support candidates for resolving the anaphora. Other works, such as that of [ORA 07], have studied the impact resolving pronominal anaphora has on automatic summarization systems. The basic idea is that the relevance of sentences is determined by the importance of the words and terms [12] it contains. Orasan and Evan's system [ORA 07] has a module dedicated to resolve pronominal anaphora by automatically identifying terms. This system gives very good results.

2.3.3. *Deep parsing algorithms*

Deep parsing approaches are based on the idea that it is necessary to use in-depth linguistic techniques that exploit the discursive structure of texts. Some of these approaches are based on rhetorical structure theory (RST) [MAN 87, MAN 88], which aims to finely exploit the structure of the discourse to generate abstracts [ONO 94, MAR 98, MAR 00b], or on meaning-text theory (MTT) [MEL 88, MEL 01].

Automatic text summarization systems based on discourse analysis arise from the idea that a text is defined by its internal structure and rhetorical relations, which depend on the language of the text. These systems give more importance to nuclear components of discursive relations (nuclearity). Using RST, [MAR 00b] divides the text into

sentence Sally arrived, but nobody saw her, the pronoun her is anaphoric, referring back to Sally. The term anaphora denotes the act of referring, whereas the word that actually does the referring is sometimes called an anaphor (or cataphor). Usually, an anaphoric expression is a proform or some other kind of deictic expression." (source: Wikipedia: http://en.wikipedia.org/wiki/Anaphora_(linguistics)).

[12] The concept of a term will be dealt with in detail in Chapter 6.

discursive units using a minimal set of relations (discourse segmentation). Once the discursive structure of the text is created, an algorithm is applied that weights and orders each element of the tree-like structure of the discourse (the higher the element in the structure, the higher its weighting). Sentences with the highest weight are selected for the summary. More or fewer elements are chosen depending on the desired length of the summary, but they always appear in the order determined by the algorithm. Recently, deep parsing approaches applied to specialized domain texts have come into existence, such as DISICOSUM [CUN 07b, CUN 08], which uses RST and specific rules to summarize texts in the biomedicine domain (see Chapter 6).

2.4. Abstract summarization

The generation of abstracts is one of automatic summarization's final objectives. Systems that produce summaries by abstraction are based on text understanding and seek to generate a grammatically correct, concise and coherent text. Very few abstract summarization systems have been created. FRUMP is one such system and was one of the first to use semantic interpretation for English texts to produce their summaries. We are, however, a long way off achieving genuine automatic text understanding. Recently, other approaches using information extraction (IE), such as Genest and Lapalme's approach [GEN 12], or the combination of IR-extraction-generation [WHI 01] (RIPTIDES), have been created. These methods will be studied in Chapter 7.

2.4.1. FRUMP *or the temptation to understand*

In 1982, Gerald Francis DeJong unveiled FRUMP [DEJ 82], a system of automatic text summarization by understanding, at Yale. FRUMP was designed to produce abstracts of documents. The main features of FRUMP were: a complex architecture aimed at "understanding" documents written in natural language and coded knowledge structures in order to "simulate" human behavior. Frump's module of understanding was conducted by a very superficial and incomplete analysis of the text. The understanding process was guided by semantics [13]. Syntax was of secondary importance.

The modular architecture of FRUMP is:

– a structure based on sketchy-scripts, an adaptation of [DEJ 79] of Schank and Abelson's [14] scripts, which exclusively contained important information related to an event:

– the *predictor* module, which searched for keywords and suggested appropriate scripts;

– the *substantiator* module, which weighted the scripts and chose one;

– and a summary generation process.

The text was interpreted through a "quick reading" (*skimming*) of the document. Sixty of so *sketchy-scripts* were produced – by hand – and related to several events. For instance: *terrorist-event, diplomatic-visit, earthquake-event*, etc. Table 2.1 shows an example of FRUMP's output, which correctly triggered the *earthquake-event* in its scripts.

However, it was not rare for FRUMP to make mistakes. An example of one of FRUMP's [15] mistakes is shown in Table 2.2.

FRUMP's main mistakes were made, firstly, due to a lack of vocabulary and, secondly, due to a lack of scripts. Other mistakes were caused by a lack of knowledge concerning syntax [SPÄ 96]. It must also be said that approximately 60 scripts are certainly not enough if the system intends to interpret documents from any area. In addition,

[13] "Semantics [...] is the study of meaning. It focuses on the relation between signifiers, like words, phrases, signs, and symbols, and what they stand for, their denotation. Linguistic semantics is the study of meaning that is used for understanding human expression through language" (source: Wikipedia: http://en.wikipedia.org/wiki/Semantics).

[14] Scripts and frames commonly found in the literature for artificial intelligence: [SCH 77].

[15] Example adapted from Horacio Saggion's, [SAG 05], http://www.macs.hw.ac.uk/esslli05/index.php.

FRUMP's knowledge had to be coded by hand, an enormous, arduous task, which, moreover, relied on human experts. This leads us to ask an obvious question: how can scripts for new domains be automatically learned?

Source	*A small earthquake shook several Southern Illinois counties Monday night, the National Earthquake Information Service in Golden, Colo., reported. Spokesman Don Finley said the quake measured 3.2 on the Richter scale, "probably not enough to do any damage or cause any injuries." The quake occurred about 7:48 p.m. CST and was centered about 30 miles east of Mount Vernon, Finlay said. It was felt in Richland, Clay, Jasper, Effington, and Marion Counties. Small earthquakes are common in the area, Finley said.*
Summary	SELECTED SKETCHY SCRIPT $EARTHQUAKE ENGLISH SUMMARY: There was an earthquake in Illinois with a 3.2 Richter scale.

Table 2.1. *Example of* FRUMP*'s output*

Source	*Vatican City. The news of the death of the Pope shakes the world. He died last Tuesday...*
Summary	Earthquake at the Vatican: a death.

Table 2.2. *Example of a* FRUMP *output error*

The significant problems with the FRUMP algorithm and the slightly more recent FASTUS system[16] [ISR 96] from the Stanford Research Institute, which used cascaded transducers, relate to the fact that there are an infinite number of topics in existence: it is incredibly difficult to introduce an exhaustive and symbolic body of knowledge into a system. In other words, the desire to build a script that can deal with each and every imaginable situation is a utopian idea.

[16] "FASTUS: A cascaded finite-state transducer for extracting information from natural-language text", http://www.ai.sri.com/natural-language/projects/fastus-schabes.html.

2.4.2. *Information extraction and abstract generation*

There are approaches, such as the RIPTIDES [WHI 01] system, which combine IE, summarization by extraction and natural language generation (NLG) to produce multidocument summaries. The RIPTIDES system is similar to Radev and McKeown's SUMMONS system [RAD 98], which summarizes articles in the field of terrorism. However, RIPTIDES was designed to summarize large documents rather than short newswires. RIPTIDES focuses more on researching recent information by generating summaries using extracted sentences.

2.5. Sentence compression and fusion

Automatic sentence compression and multisentence fusion (MSF) are two relatively new subjects of research in automatic text summarization. Understanding and mastering these topics is important, as they enable a number of improvements to be made, including reducing redundancy and creating summaries that are more similar to human abstracts. Nevertheless, sentence compression and sentence fusion are entirely different subjects (see sections 7.6 and 7.7).

2.5.1. *Sentence compression*

The idea behind sentence compression is simple: to eliminate all the non-essential information in a sentence while preserving grammaticality. Turning this idea into an algorithm is, however, a whole other matter, and no method has ever found a definitive solution to this problem.

There is a range of applications for sentence compression: automatic title generation, generating subtitles for the media, displaying information on mobile devices and so on. Automatic summarization is also concerned with sentence compression. In fact, sentence compression is considered a resource for optimizing automatic summarization [MOL 10a], but also a sort of summary itself, in which the compression rate can reach 33% [YOU 06]. A summary generated by sentence compression is one in which each sentence from the original document has been compressed. Sentences are never deleted [YOU 07].

There is a general consensus that sentence fusion and sentence compression are basic forms of paraphrasing. They are, therefore, steps toward abstract generation and crucial for improving the quality of this type of summary. How can sentence fusion and sentence compression be modeled while attempting to preserve semantics and grammaticality? Since Knight and Marcu's [KNI 00, MAR 00b, KNI 02] founding works, these tasks have been tackled using two methods: symbolic approaches, see, for example, [JIN 00a, GAG 06, YOU 07], and statistical approaches. These algorithms will be studied in section 7.7.

2.5.2. *Multisentence fusion*

The idea of MSF or multisentence compression (MSC) arises from multidocument summarization: the more documents there are about a certain subject, the more important the redundancy in the generated extract. The aim is, therefore, to reduce redundancy while maintaining the essential information contained in a group of semantically similar sentences. Barzilay and McKeown introduced this interesting idea in 2005 [BAR 05] in the context of multidocument summarization: a set of sentences (extracted according to their weighting) from different documents is fused and compressed according to similarity. Given that MSF is fundamentally paraphrasing, it can also be used in the context of abstract generation. In fact, one form of basic paraphrasing is the fusion of similar sentences.

Graph-based algorithms are well-adapted to the task of MSF [FIL 10, BOU 13, TZO 14]. MSF algorithms will be studied in more detail in section 7.6.

2.6. The limits of extraction

The extraction approach has become a paradigm for generating summaries for several reasons: it is simple, easy to implement, has a modular architecture, is robust and performs well. Nevertheless, despite countless successes, extraction does have several problems that are proving difficult to resolve, namely, the cohesion and coherence of summaries and the fact that the performance of extraction systems levels off after a certain period of time.

2.6.1. *Cohesion and coherence*

Extracted sentences are often incoherent and difficult to read. Cohesion and coherence are two ways of directly evaluating synthetically [17] produced summaries, as they are strongly connected to the quality of the summary. According to Barzilay and Elhadad [BAR 97]:

> "*Cohesion is a device for 'sticking together' different parts of the text. Cohesion is achieved through the use of semantically related terms, co-reference, ellipsis and conjunctions*".

Cohesion is located at the linguistic level of the sentence, whereas coherence is located at the higher level of semantics. Although cohesion and coherence are easily identifiable when a human reads a summary, automating the task is a whole other matter.

DEFINITION 2.1.– *Cohesion between the sentences of a text is related to the absence of anaphora and unresolved temporal references in the generated summary.*

For instance, let us consider the following sentences:

1) *I love cherries.*

2) *This summer they are, however, too acid for my taste.*

The two sentences are linked by the anaphora (rhetorical connector) "they" in sentence number (2). But, if the second sentence is preserved

[17] Or intrinsically. Cohesion and coherence will be discussed again in Chapter 8.

in an extract – and not the first – it risks compromising the cohesion of the summary, given that it is denied the context (what does “they” refer to?) that is established in sentence number (1) (“[the] cherries”). A simplistic solution would be to add as many sentences as necessary to produce a cleaner summary. However, this would not be possible given that the anaphora may be at some distance from the sentence in question (several sentences before or after). A definitive solution would be to identify and resolve the anaphora present in a text. This is, however, an extremely difficult task that has proved to be much too dependent on the language[18].

DEFINITION 2.2.– *The coherence of the summary is the absence of contradictions and/or redundancy in the sequence of sentences in a document.*

Let us consider, for example, a document and its summary that were produced by an extraction algorithm (shown in Table 2.3). It is immediately clear that coherence has not been respected in the summary. However, automatically identifying and resolving the lack of coherence and/or cohesion in summaries is a difficult task and falls outside the scope of this work.

A simple hypothesis for improving the coherence of summaries produced by extraction is that it is preferable to cover fewer subjects, and cover them in-depth, than to cover too many. This approach was followed by White *et al.* [WHI 01]. The approach favors sentences with high scores but does not neglect sentences with lower scores. The system attributes a score to a set of sentences by adding up the weights of sentences with better scores and the weights of some adjacent sentences, such as preceding sentences with an initial pronoun or strong discursive connectors: “such as”, “however”, “all the same”, etc.

[18] Resolving anaphora is an extremely difficult NLP task that is beyond the scope of this work. For more information on algorithms for resolving anaphora, see [SAF 96] and [ROG 01].

Source	The aircraft manufacturing industry is still in crisis in 2011. In 2010 Airbus faced many problems due to financial difficulties and was forced to halt production of its new long-haul carrier A380. Transatlantic, its Boeing rival, has finally started production of its B770. The new B770 has received a large number of pre-orders from Asian airline companies. This is good news for the manufacturer who may end up owning a large share of the aircraft industry.
Summary	In 2010 Airbus faced many problems due to financial difficulties and was forced to halt production of its new long-haul carrier A380. This is good news for the manufacturer who may end up owning a large share of the aircraft industry.

Table 2.3. *Example of the loss of coherence in extract summarization*

2.6.2. *The HexTAC experiment*

Genest *et al.*'s 2009 study "The Creation of a Manual Extractive Run", which looked at human extract generation at the Text Analysis Conference 2009 (TAC'09) (main and update) task[19], gave worrying results for system developers. This was the HexTAC[20] campaign [GEN 09a]. Multidocument and update extracts were created by humans. Their linguistic quality and global scores were then compared to the best system and to abstracts created by humans. The results of this study are shown in Table 2.4.

Manual evaluations revealed that humans produced better summaries than any other system that participated, even when limited to extraction. That said, people who were free to generate real abstracts outdid the best human extracts. The significant gap between the global score for human extracts and the global score for human abstracts emphasizes the limits of extractive methods, which are used by the majority of systems. Extractive methods are, in fact, used by the best automatic summarization systems, but they will never achieve the same

[19] See the website of Text Analysis Conference 2009 (TAC): http://www.nist.gov/tac/2009/Summarization/ and section 8.6.

[20] HUMAN EXTRACTION FOR TAC. For articles, resources and the HexTAC application (available under GPLv3 license) visit the RALI website: http://rali.iro.umontreal.ca/rali/?q=fr/hextac.

quality as human writers. Therefore, research must be directed toward finding solutions by abstraction rather than by extraction, which is known to be limited [GEN 09a, NEN 11].

Main Task	**Linguistic Quality**	**Global Score**
Human Abstracts	**8.915**	**8.830**
Human Extracts	7.477	6.341
Best system (ICSI)	*5.932*	*5.159*

Update Task	**Linguistic Quality**	**Global Score**
Human Abstracts	**8.807**	**8.506**
Human Extracts	7.250	6.114
Best system (ICSI)	*5.866*	*5.023*

Table 2.4. *Human performance* vs *systems performance at TAC'09 [GEN 09a]. The system ICSI will be presented in section 4.6.4*

2.7. The evolution of automatic text summarization tasks

Automatic text summarization techniques have slowly evolved out of the issues that researchers have been trying to resolve. Since the pioneering works of the 1950s, automatic text summarization tasks have, therefore, moved toward resolving increasingly complex issues to meet the ever-growing list of user expectations. This section will describe various traditional and recent summarization tasks and lay the foundations for what automatic text summarization could become in the near future.

2.7.1. *Traditional tasks*

There are countless traditional variants of automatic text summarization: generic summarization, guided single-document, multidocument and multilingual summarization and various combinations of the above. Each task has different problems and requires different approaches to tackle them.

Single-document summarization

Single-document summarization is the simplest variation of text summarization. It consists of producing a summary that preserves the most important information contained in the source document. Though it may appear the simplest of tasks, it still faces countless difficulties. In fact, the performance of the system depends on the type of document to be summarized. For instance, while it is possible to generate a more or less coherent summary of journalistic articles, it is currently pretty much impossible to generate a relevant summary from literary documents [CEY 09]. Moreover, since the summary is produced from sentences extracted from the document, correctly assembling the sentences is essential if cohesion and coherence are to be achieved.

Guided summarization

In contrast to generic summarization, oriented summarization (also known as guided or personalized summarization) consists of producing a summary that meets a user's information needs. Normally, these needs are expressed via a request (query). This query must enable the system to isolate parts of the document that deal with one or several specific topics. The aim is to produce a summary that only includes passages that deal with the information the user requested.

Multidocument summarization

Multidocument summarization can be applied to generic and oriented summarization. Multidocument summarization involves generating a relevant summary from a set of heterogeneous (though not necessarily relevant) documents. It is impossible to manually process documents once a certain volume is reached. Online press dispatch aggregators, such as the COLUMBIA NEWSBLASTER[21], sought inspiration from research into multidocument summarization. Introducing multidocument into the summarization task causes new difficulties: sentences extracted from different texts may compliment or contradict each other, and there may also be repetitions. Information

[21] This system will be explained in section 5.5.

redundancy between sentences is a major issue. In fact, it is even trickier to manage cohesion, coherence and redundancy in multidocument summaries than in single-document summaries.

Multilingual summarization

Automatic summarization was initially monolingual: the source and summary were written in the same language. However, it did not take long for the question of multilingualism to be tackled. The advent of the Internet meant that, when researching information, it was not unusual to find texts to read in different languages. When automatic translation techniques were developed, it became possible to produce a summary in a different language to that of the source documents. The use of robust statistical techniques enables methods for sentence weighting to be language-independent. SUMMARIST [HOV 98, LIN 98, HOV 99], one of the first multilingual systems, aims to automatically produce summaries from a set of documents written in several languages[22]. To produce a multilingual multidocument summary, sentences (written in different languages) are extracted, assembled and finally translated into the desired language. In addition to the aforementioned problems, the question of the quality of the translation must also be taken into account.

2.7.2. *Current and future problems*

New online text contents have created new automatic summarization tasks: torrents of information, discussion threads (blogs), email summarization, etc. Specialized domain texts (medical, legal etc.) also cause particular difficulties, due to the specificity of their terminology and structure. Automatic summarization algorithms must, therefore, adapt and evolve in order to be able to process these diverse sources of information.

[22] Documents in English, Japanese, Spanish, Arabic, Indonesian and Korean from http://www.isi.edu/natural-language/projects/SUMMARIST.html.

Recently, automatic summarization has started to use semantic resources such as the lexical databases *WordNet* for English[23] and *EuroWordNet* (EWN) for a variety of other languages[24] (see section 6.3.1). This enables algorithms for sentence weighting to be guided by the word frequency in a document as well as the term frequency and related terms: networks of synonyms, hyponyms, hyperonyms, semantically close terms etc. Opinion summarization presents a significant challenge for the community. The multilingual task at the Text Analysis Conference (TAC) 2011 organized by the National Institute of Standards and Technology (NIST) introduced a new area of research. The NIST's objective is to promote the use of multilingual algorithms in summarization. This involves making algorithms or sets of monolingual resources multilingual. Finally, users are increasingly calling for multimedia summarization and abstract (in addition to extract) generation.

Summarization based on the source of the document

Until the present day, the majority of summarization algorithms have targeted journalistic articles and scientific documents. There are, however, works in existence that deal with different types of documents. A wide range of problems has been tackled including the summarization of email threads [WAN 04], online discussions [ZHO 05a], man–machine oral communication [GAL 06], literary works [CEY 09], blogs [HU 07], web pages [BER 00] among others.

Specialized-domain summarization

In contrast to journalistic articles (that benefit from large amounts of data made available through evaluation campaigns), there is very little or no data available about other types of document. This is certainly true for the automatic summarization of specialized domains, such as organic chemistry, biomedicine or legal decisions [FAR 04a].

[23] "WordNet: a Lexical database for English", website: http://wordnet.princeton.edu.

[24] EuroWordNet is a multilingual database for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). Website: http://www.illc.uva.nl/EuroWordNet/.

These types of documents have a very specific structure and use specialist terminology [VIV 10a]. To resolve this absence of reference evaluation data, the scientific community has created specific evaluation *corpus* [BOU 08c].

Update summarization

Introduced during the Document Understanding Conference (DUC) 2007 evaluation campaign (see section 8.6), update summarization aims to improve summary quality by taking into account the user's expectations and previous knowledge. Update summarization asks the following question: has the user already read documents or summaries about the topic in question? If the answer is yes, producing a summary that only contains new facts could be extremely useful. There is one serious challenge: reducing information redundancy for documents the user has already read.

Sentence compression and multi-sentence fusion

Automatic sentence compression and MSF are a recent addition to research in automatic summarization. In fact, they can be considered both as a type of summary and as a complementary task to extract summarization. Automatic sentence compression consists of eliminating the non-essential elements in a sentence [YOU 06]. Jing's research [JIN 00b] into the different actions employed by professional summarizers shows that sentence compression is an essential part of generating abstracts. The process also involves sentence fusion, transforming syntax and selecting sentences. Sentence compression can, however, be considered as one type of generic summary, which contains a simplified version of each sentence. MSF consists of generating a sentence from the fusion of a group of k related sentences [BAR 05, FIL 10, BOU 13, TZO 14]. Thus, the main objective is to reduce redundancy, while maintaining the essential information contained in a group of sentences.

Semantic summarization

Automatical text summarization using semantic resources is an active area of research particularly in specialized domain summarization. Plaza *et al.* [PLA 10] developed a method for summarizing concepts based on biomedical documents. They show that the performance of the system improved with the use of disambiguation. The system represents documents as graphs that are made up of concepts and relations. Vivaldi *et al.* [VIV 10a] developed an automatic summarization algorithm for specialized domain documents that combined terminological and semantic (a term extractor and an ontology) resources. The term extractor locates terms according to their termhood[25] in the documents, and an ontology enables the semantic similarity between terms in the *corpus* and terms in the title to be calculated so that sentences can be weighted.

Opinion summarization

This task differs from traditional text summarization, as it processes customers' opinions about a certain product's features or aspects. The opinions expressed can be positive, negative or neutral. Opinion summarization does not involve selecting a subset of comments or rewriting certain sentences to show the main ideas, like in traditional text summarization; rather it involves assembling the positive or neutral opinions expressed about a product in a coherent way. Several solutions are centered around semantic resources.

Multi and cross-lingual summarization

The TAC 2011 MultiLing Pilot task aimed to evaluate the (partial or complete) application of language-independent summarization algorithms for a variety of languages. Each participant provided summaries in seven different languages (Arabic, Czech, English, French, Hebrew, Hindi and Greek) based on the corresponding *corpora*. The systems taking part applied their methods to at least two languages. Systems that applied their methods to other languages were favored. The Multiling Pilot task required participants to generate a unique, fluid and representative summary of a *corpus* that described a sequence of events (a sequence of events is an independent and temporal set of descriptions of "atomic events"). Each event in the corpus contained at least three distinct events. The summary (between

[25] See section 6.3.1.

240 and 250 words in length) was produced in the same language as the source documents[26]. This campaign was followed by MultiLing 2013[27], which focused on three key aspects: multilingual multidocument summarization, multilingual summary evaluation and multilingual summarization data collection and exploitation. Users who only speak one language, therefore, require multi- and cross-lingual summarization and research to be able to make use of collections of documents available in other languages.

Ultra-summarization

Summarizing short texts or ultra-summarization is useful for mobile devices and platforms that have limited space (reading online news on a mobile phone) as well as for the automatic generation of newspaper headlines, and keyword and title generation [BAN 00]. Ultra-summarization can become oriented and cross-lingual.

Tweet summarization (short texts in microblogs)

Tweet summarization does not involve summarizing tweets, which are already too short, but summarizing a set of tweets. Multitweet summarization quickly enables users to understand the main message of a large number of tweets about a more or less homogenous topic [CHA 11, LIU 12]. This task can be combined with opinion summarization.

Multimedia summarization

Multimedia summarization combines information sources available online: text documents, images, audio and video. Processing these texts remains a difficult task and very few systems capable of doing so currently exist. Nevertheless, it is becoming an increasingly active area of research in the domain of automatic summarization, see, for instance, Hori and Riedhammer *et al.*'s work [HOR 02, RIE 10] on

[26] Source: “Multi-Lingual Summarization in the Text Analysis Conference (TAC) 2011”, http://users.iit.demokritos.gr/ggianna/TAC2011/MultiLing2011.html.

[27] http://multiling.iit.demokritos.gr/pages/view/662/multiling-2013.

speech summarization[28]. There is a little doubt that multimedia summarization will become a major issue in years to come.

Abstract generation

Extracted sentences are grammatically correct, but they do lack coherence and cohesion and sometimes include unresolved anaphoric references [GEN 10]. Users are increasingly calling for researchers to break away from extraction and move toward *real* abstraction, however, this is not an easy task within the paradigm of pure extraction. The addition of advanced simplification methods, context compression, sentence fusion and text generation should probably be considered for abstract generation. Some of these methods will be reviewed in Chapter 7.

2.8. Evaluating summaries

Evaluating automatically generated summaries has always been an extremely difficult task. It is an open problem to which the scientific community has responded with partial solutions. Evaluation methods, such as precision and recall, which are used a great deal in IR, are not well-adapted to this task, given that what goes in and comes out of automatic summarization systems is natural language, which varies considerably. There are two prevalent types of evaluation: intrinsic evaluation (about the summary itself) and extrinsic evaluation (measuring the quality of the summary through other NLP tasks). This work will look at intrinsic evaluation.

One of the first serious studies in this area was conducted by Radev *et al.* in 2003 [RAD 03]. They developed a set of automatic measures (cosine, precision/recall, longest common subsequence (LCS) etc.) that were evaluated on a large bilingual (English/Chinese) corpus.

[28] In France, the *Agence Nationale de la Recherche* (ANR) launched the "Multimedia, multi-document and multi-opinion" (RPM2) project in 2007 "[this] project aimed to develop multi-document summarization methods for text, audio and video as well as for multimedia" (source: http://www.rpm2.org/).

Evaluation methods that measure linguistic quality (such as readability, grammaticality etc.) or that focus on the content (such as informativity), in other words, the quantity of essential information in a summary, have been developed. Semi-automatic evaluation systems, such as PYRAMID [NEN 04], gave priority to these approaches in the DUC and TAC campaigns. As a result, methods that compare the number of n-grams in a candidate summary and in reference summaries, for instance, ROUGE [LIN 04], were developed. Finally, some other methods that avoid the use of reference summaries have recently appeared [LOU 08, SAG 10, LOU 13]. In the following chapters, which will discuss evaluating the quality of summaries, reference will be made to these evaluation metrics. All methods have advantages and disadvantages. A series of approaches for evaluating summaries will be reviewed in Chapter 8.

2.9. Conclusion

A number of factors influence automatic text summarization: the type, the genre, the aim, the quantity and the heterogeneity of the sources, user expectation, the compression rate and so on. Algorithms for and systems of automatic text summarization are designed in accordance with these sometimes conflicting constraints. There are three broad approaches to automatic text summarization: algorithms based on text understanding, algorithms based on extraction and sentence compression methods. Though appealing, methods based on a deep understanding of the text face difficulties, namely, the ability to appropriately code an exhaustive body of knowledge. However, extraction approaches (using either surface-level, intermediate-level or deep parsing analysis), although simplistic, are easier to implement in real systems. Cremmins's work [CRE 93, CRE 96] has shown that, to be able to produce a summary, professional summarizers do not really need to understand the text: in most cases, a quick and superficial reading will suffice. According to this hypothesis, a machine can attempt to replicate this task, without ever having to really understand the text. Algorithms that summarize by extraction are attempting to do exactly this, and they have enjoyed great success in recent years. Nevertheless, problems related to the coherence and cohesion of summaries continue to blight extraction systems. The scientific community has not managed to resolve these issues completely, though several partial solutions have been developed. Two possibilities are the reformulation and fusion of extracted sentences. Sentence compression is a step toward reformulation, but more research needs to be carried out in this area. Current abstraction systems, based on IR techniques, partial understanding and NLG are trying to produce abstracts and avoid the problems of cohesion and coherence.

Automatic text summarization comprises several traditional tasks (generic single-document summarization, generic or guided multidocument summarization and multilingual summarization). A great deal of research is being carried out with regard to contemporary tasks (specialized-domain summarization, update summarization, sentence compression and summary evaluation). New trends (semantic and terminological summarization, opinion summarization, ultra-summarization, TAC Multiling and Multiling-ACL pilot tasks and abstract generation) present a new set of challenges to be overcome. Another fashionable task is currently being developed: multimedia summarization, which summarizes heterogeneous documents that contain text, audio, images and video.

3

Single-document Summarization

The most simple task of automatic summarization is generic single-document summarization. Introduced in Hans Peter Luhn's[1] 1958 work in the form of a summarization system by extraction [LUH 58], this seemingly simple task is still far from being completely solved. The overload of textual information on the Internet has re-sparked users' interest in this type of summary. Single-document summarization transforms a source text into a condensed, shorter text in which the relevant information is preserved. Several extraction algorithms for sentence weighting (statistical methods, graph-based, using rhetorical structure, etc.) have been created. This chapter will present the foundations of single-document automatic summarization systems. Founding and current approaches will be described, as well as the problems which exist with regard to the production of single-document summarization by sentence extraction.

3.1. Historical approaches

Ann Mathis' technical report[2] presented state-of-the-art algorithms for automatic summarization from its invention in 1958 until 1972

[1] Hans Peter Luhn, born July 1, 1896, in Wuppertal (Germany) and died August 19, 1964, was a computer scientist who worked for IBM. He was the creator of the Luhn formula modulus 10 (which is still used to verify credit card digits), the KWIC (Key Words in Context) concordance method and authored the founding works in document indexing and automatic text summarization. He developed several statistical methods whose occurrences were used to select words or their stems in natural language. See Claire K. Schultz, H.P. Luhn: pioneer of information science: selected works, 1968, Spartan Books, http://faculty.libsci.sc.edu/bob/ISP/luhn.htm, and http://en.wikipedia.org/wiki/Hans_Peter_Luhn.

[2] Available from the Institute of Education Sciences (ERIC): http://files.eric.ed.gov/fulltext/ED071686.pdf.

[MAT 72]. It is an illustrative study as it gives an account of the problems posed by automatic text summarization as well as the solutions researchers were proposing at this time. Many ideas included in the study are still the pillars of modern research into automatic text summarization. The emergence of ideas relating to statistical extraction, preprocessing, methods using linguistic or semantic approaches and even ways of evaluating systems is quite remarkable. The objective at the time was clear: to automatically generate abstracts with a machine. To this end, several sentence weighting, sentence compression and fusion strategies were developed and evaluated in the domain of chemistry texts.

Mathis' [MAT 72] report studied the following algorithms:

– 1958: the Luhn algorithm [LUH 58];

– 1959: the Oswald study;

– 1960–1961: the ACSI-Matic study conducted by IBM;

– 1961–1962: word-association research;

– 1961–1969: the TRW studies conducted by Edmundson and Wyllys [EDM 61, EDM 69];

– 1964-1971: the Earl study;

– 1968-1971: the Soviet study of Skoroxod'ko;

– 1971-1972: the Rush study [RUS 71].

A brief overview of this report will be given. The following section will describe three methods in more detail: the Luhn algorithm, Edmundson's studies and the Rush study.

Luhn developed the first algorithm for automatic summarization based on the word frequency. The Oswald study[3] used an indexing criterion to select sentences to produce a summary. The stages involved in the Oswald algorithm are as follows: (1) the number of significant

[3] Edmundson H.P., Oswald V.A., Wyllys R.E., "Automatic indexing and abstracting of the contents of documents", PRC-R-126 Planning Research Corp., Los Angeles, CA, Prepared for Rome Air Development Center, Air Research and Development Command, Griffiss Air Force Base, New York, 31 October 1959.

words is determined in a similar way to that used by Luhn. (2) The non-significant words with an occurrence number greater than 1 are identified. The juxtaposition of these words constitutes a *multiterm*. (3) Sentences containing two or more *multiterms* are identified. These sentences are ranked according to the number of *multiterms* and the highest ranking sentences are finally extracted in accordance with the desired compression rate. Oswald did not have a computer at the time, so he simulated the algorithm manually. However, this study is interesting for two reasons: the relation between an index and a summary is recognized as is the use of multiterms (chains of words with a length greater than 1) in the sentence weighting.

ACSI-Matic, conducted by IBM[4] for the Army Department's Assistant Chief of Staff for Intelligence, was the only study at the time which led to an abstraction system being part of an operational information system. The ACSI-Matic software carried out the following processing operations: (1) establishing the Luhn technique for sentence weighting, (2) special processing of documents with an unusual and high quantity of rare words, (3) special processing of documents with an extraordinary number of long sentences, (4) selecting weighted sentences to form a provisional summary, and (5) reducing the redundancy of selected sentences.

Word association studies have highlighted the importance of word clusters in written discourse. Doyle *et al.* [DOY 61, DOY 62] attempted to establish a measure of semantic information content (i.e. measuring the value of certain text structures in accordance with the author's intention), which used language representations. Doyle's concept of association maps is based on statistical association criteria. The generation of these maps requires a correlation matrix, $n \times n$ (n = number of keywords to be correlated), which is calculated through the Pearson correlation coefficient. At that time, no concerted effort was made to use semantic structure to produce summaries. But given that

[4] IBM Corp., Advanced Systems Development Division, ACSI-Matic Auto-Abstracting Project, Final Report, vol. 1 and 3, Yorktown Heights, New York, 1960-1961.

summarization is fundamentally a question of abstracting the semantic content, these authors concluded that research into semantics would be useful for the production of abstracts.

The research into abstracts and informative extracts conducted by Earl [DOL 65] at Lockheed Missiles and Space Co. studied the role of morphology, phonetics and syntax in summaries. This study, supported by the Office of Naval Research, considered linguistics a precondition for the automatic production of abstracts. Resources relating to part-of-speech tagging (POS tagging or POST, see Appendix A.1.5) were generated. They were used to determine the degree of proximity between sentences and thereby to produce an abstract. Although interesting experimental data were obtained, Earl *et al.* did not really succeed in constructing an automatic abstraction system.

E.F. Skoroxod'ko led research on automatic summarization at the Academy of Sciences of the Ukrainian SSR in Kiev (ex-USSR). His research [SKO 68, SKO 71] was based on the hypothesis that, in order to obtain correct and stable results, the automatic abstraction method must adapt itself to the source. As a consequence, it was presumed that the summary depends to a great extent on the characteristics of the text itself. The individual characteristics of a text were modeled as a semantic network: a graph where the nodes are the sentences, connected by edges (their semantic relations). A and B are two sentences and "a" $\in$ A and "b" $\in$ B are two words. A and B are semantically linked if (1) at least one noun appears in the two sentences, (2) if "a" and "b" have been predefined as being semantically close or (3) if "a" and "b" are linked in relation to another given text. The importance of a sentence is directly proportionate to the number of sentences which are semantically linked to it. This approach strongly relies on the co-occurrence of words and the correspondence of words and synonyms in a manually constructed dictionary. Skoroxod'ko calculated a sentence weighting function based on the connectivity of the graph. This function enabled him to establish four types of graph: chained, ringed, monolithic and piecewise. On the basis of this measure, Skoroxod'ko concluded that it is impossible to form an adequate extract of sentences in chained or ringed graphs because all the sentences have more or less the same semantic value. Summaries can be generated from texts when the semantic relations can be represented as either monolithic structures or in piecewise. These were pioneering ideas with regard to the use of linguistic, semantic and graph resources applied to extractive automatic summarization.

3.1.1. *H.P. Luhn's Automatic Creation of Literature Abstracts*

> Abstract: Excerpts of technical papers and magazine articles that serve the purposes of conventional abstracts have been created entirely by automatic means. In the exploratory research described, the complete text of an article in machine-readable form is scanned by an IBM 704 data-processing machine and analyzed in accordance with a standard program. Statistical information derived from word frequency and distribution is used by the machine to compute a relative measure of significance, first for individual words and then for sentences. Sentences scoring highest in significance are extracted and printed out to become the "auto-abstract".
>
> H.P. Luhn, The Automatic Creation of Literature Abstracts

Hans Peter Luhn was the pioneer of the creation of algorithms for automatic text summarization. In his work (December 1957–April 1958)[5] [LUH 58], he described some of the advantages of automatically created summaries over manual summaries: a low production cost and a reduction of problems relating to subjectivity and variability, which were observed in summaries generated by professional summarizers. H.P. Luhn was already worried about the problem of the overload of information, which was visionary and very much ahead of his time. In 1958 at IBM, Luhn developed a simple and specific algorithm for scientific and technical articles, which used the distribution of the word frequency to weight sentences (see Figure 3.1).

In Luhn's own words, his method was miles away from text understanding and even linguistics: "As pointed out earlier, the method to be developed here is a probabilistic one based on the physical

[5] Presented at IRE National Convention, New York, 24 March 1958.

properties of written texts. No consideration is to be given to the meaning of words or the arguments expressed by word combinations". Indeed, Luhn based his algorithm on the use of repeated words: "The justification of measuring word significance by use frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject. This means of emphasis is taken as an indicator of significance. The more often certain words are found in each other's company within a sentence, the more significance may be attributed to each of these words" [LUH 58].

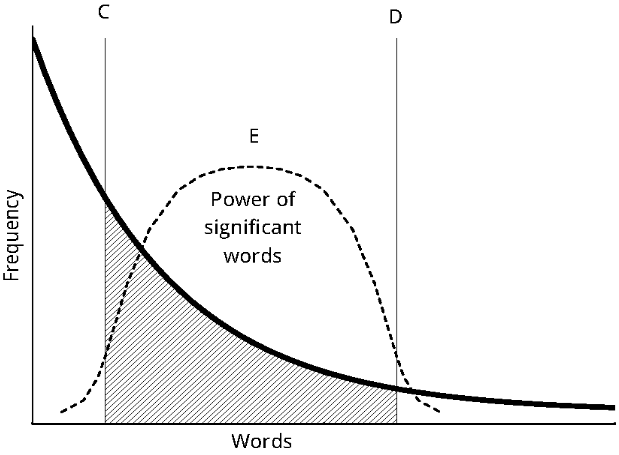


Figure 3.1. *Distribution of the word frequency where the zone included between C and D constitutes an interesting zone of discrimination, according to [LUH 58]*

Luhn also considered problems caused by the variability of words. He used normalization, which grouped similar words from the perspective of spelling. The aim of normalization (Luhn was, in fact, thinking about what we now call *stemming*)[6] was to liberate words from orthographic variations in order to fuse words carrying the same meaning by grouping them into the same lexical family. According to

[6] See Appendix A.1.2 for more details concerning the process of word normalization.

Figure 3.1, the word frequency has an impact on their discriminatory power. The words included between points C and D, whose frequency is neither too high nor too low, constitute an interesting zone of discrimination, modeled in the shape of a bell with a maximum point located at E. This set of significant keywords has therefore been used to weight sentences in scientific articles for their extraction.

3.1.2. *The Luhn algorithm*

The Luhn algorithm is composed of two phases: first the source document is preprocessed and then the sentences s are weighted according to the occurrence of words and keywords. This is illustrated in Figure 3.2. Luhn programmed his algorithm on an IBM 704 machine[7] (available programming languages were Fortran and LISP).

3.1.2.1. *Preprocessing*

The algorithm starts by filtering common words (function words) with the help of an antidictionary[8] (stoplist). This list was created manually and contains pronouns, articles and prepositions in English. Other words are statistically grouped according to similar spelling: words with a common prefix and less than 6 different characters are considered to belong to the same lexical family. Then the occurrence number of each lexical group is counted. The infrequent words are eliminated and a small list of frequent words, which are not function

[7] The 704 series was made famous by an anecdote about science fiction: “One of the more famous moments in Bell Labs’ synthetic speech research was the sample created by John L. Kelly in 1962, using an IBM 704 computer. Kelly’s vocoder synthesizer recreated the song ‘Bicycle Built for Two’, with musical accompaniment from Max Mathews. Arthur C. Clarke, then visiting friend and colleague John Pierce at the Bell Labs Murray Hill facility, saw this remarkable demonstration and later used it in the climactic scene of his novel and screenplay for *2001: A Space Odyssey* where the HAL9000 computer sings this song as he is disassembled by astronaut Dave Bowman”. Sources: http://www3.alcatel-lucent.com/ and https://en.wikipedia.org/wiki/John_Larry_Kelly,_Jr.

[8] An antidictionary or stoplist contains a list of words which are undesirable for automatic processing. See Appendix A.1.2 for other lexical filtering and text normalization processes.

words, is obtained. These words are considered to be significant (keywords) with regard to the topic. Sentences are weighted according to the number and the proximity of their keywords.

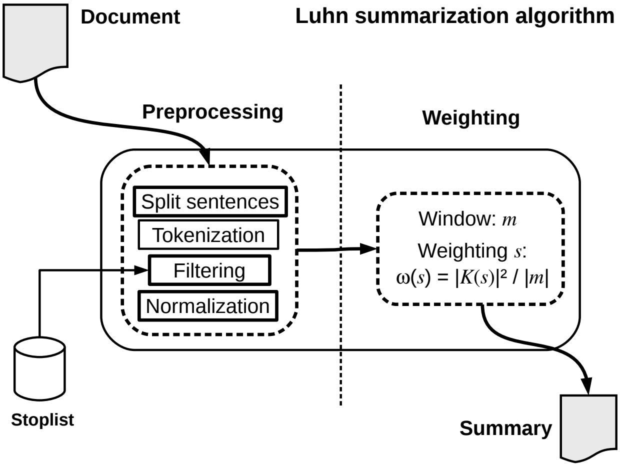


Figure 3.2. *The Luhn algorithm*

3.1.2.2. *Sentence weighting*

The Luhn algorithm (Figure 3.3) for sentence weighting is as follows:

Constructing the window: a window of the size m words is defined as the longest segment in a sentence between two keywords. In Figure 3.3, this window is shown between the square brackets and contains $K(s) = 4$ significant keywords, which are shown as ●. Attention must be paid to have not more than four non-significant words outside this window.

Sentence weighting: the weight $\omega(s)$ of a sentence s is determined by calculating the ratio between the square of the number of keywords $K(s)$ and the number of words inside the window $|m|$:

$$\omega(s) = \frac{|K(s)|^2}{|m|} \qquad [3.1]$$

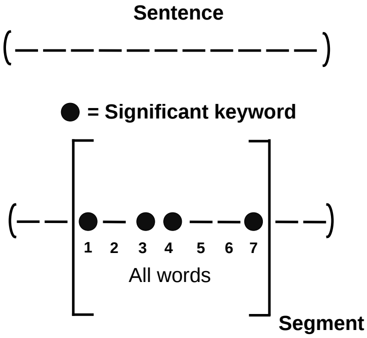


Figure 3.3. *The Luhn algorithm for sentence weighting using significant keywords ● inside a window*

If this value is high (relevant or maximal) in relation to other weightings, the whole sentence s will be extracted. In the example in Figure 3.3, the weight of the sentence s is $\omega(s) = 4^2/7 = 2.3$.

Of course, one problem is being able to objectively determine the points C and D (zone of discrimination in Figure 3.1). Luhn carried out a limited and qualitative empirical evaluation of his system by comparing the produced summaries with a direct reading. Despite this limited evaluation, his statistical idea for automatically producing text summaries had a major impact on the domain: most current systems are based on this idea.

For Luhn, the automatic production of abstracts represented great progress. In his own words: “If machines can perform satisfactorily within the range outlined in this paper, a substantial and worthwhile saving in human effort will have been realized. The auto-abstract is perhaps the first example of a machine-generated equivalent of a completely intellectual task in the field of literature evaluation” [LUH 58].

3.1.3. *Edmundson’s linear combination*

Studies conducted at the Thompson Ramo Wooldridge, Inc. (TRW) had two objectives: to develop an abstract system to produce indicative summaries and a methodology of new efficient criteria to generate summaries. In 1961, Harold Parkins Edmundson and Roland Eugene Wyllys proposed several characteristics (features) for giving a score to sentences in a *corpus* in order to rank them [EDM 61]. The sentences with the best scores were preserved to produce a summary. Relative word frequency was used to get closer to the meaning of words, groups of words and sentences. The features used were: the position of the sentence (in a journalistic article, for example, the first sentences are the most important), proper nouns, keywords in the title and the length of sentences.

In 1969, Edmundson *et al.* built on Luhn's work by using the presence of *cue-words* such as "significant", "impossible", "hardly", etc., the position of sentences and the presence of words linked to the structure of the documents (words in the title, subtitles and sections) [EDM 69]. To improve the relevance of a summary, it is therefore essential to have a list of domain-specific cue-words. But, given that Edmundson wanted the most generic system possible, he used a set of manually chosen cue-words indicating a certain importance, independent of document type: "essential", "fundamental", etc. He therefore constructed three cue-word dictionaries:

– *bonus words*, 783 positively relevant words;

– *malus words*, 73 negatively relevant words;

– *nulls words*, 139 irrelevant words.

Edmundson's algorithm

Edmundson developed the following algorithm:

– set the parameters $\lambda_\bullet$ at integer values and an empirical length parameter l. The weight $\omega(s)$ of a sentence s was calculated via a linear combination:

$$\omega(s) = \lambda_C C(s) + \lambda_K K(s) + \lambda_T T(s) + \lambda_L L(s) \qquad [3.2]$$

where C are the cue-words, K are the keywords, T are the words in the title and L is the position of the sentence;

– sort the sentences in decreasing order according to their weight $\omega(s)$;

– choose sentences whose ranking is lower than the length of the parameter l;

– the summary is generated by displaying the title, authors and extracted sentences.

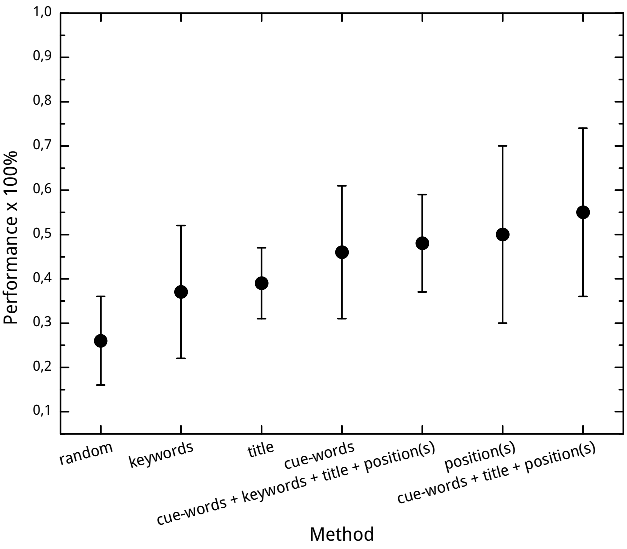


Figure 3.4. *Performance of the combination of features proposed by Edmundson [EDM 69]*

This approach was manually evaluated by Edmundson, by comparing summaries produced by his system with reference summaries (sentences extracted manually and random extracts). The *corpora* that Edmundson used had the following order of magnitude: a learning *corpus* composed of 200 reports about physics, life sciences, information sciences and the humanities. The test *corpus* was composed of 200 documents about chemistry with text lengths of between 100 and 3,900 words. Edmundson showed that the combination of cue-words + title + position(s) performed better than the distribution of the frequency of keywords alone. He also found that the most important parameter was the position(s) of the sentence in the document (see Figure 3.4).

Figure 3.4 must be interpreted as the average performance of combinations and their standard deviation. One of the main disadvantages of this method is that cue-words depend on the genre of the document, as pointed out by [LIN 97]. It must be said that Edmundson was the first person to introduce, in the evaluation of automatic summaries, a concept which would, years later, be called baseline (see section 8.3.1). In Figure 3.4, the average performance of the random baseline is approximately 25%.

3.1.4. *Extracts by elimination*

The elimination of sentences was introduced for the first time by Rush *et al.* in 1971 [RUS 71], and was then improved by J.J. Pollock and Antonio Zamora [POL 75] [9]. These authors developed ADAM, an algorithm for sentence rejection rather than selection. This algorithm differs from other algorithms because the task it accomplishes is very specific: the creation of summaries from scientific articles in the domain of chemistry. The advantages of specializing a summarization system to a predetermined type of document while exploiting this additional knowledge were thus presented. The algorithm is based on the presence of keywords (which were very similar to Edmundson's criterion), the list of which, however, was specific to chemistry. In particular, they were interested in the subdomains of chemistry: each subdomain had a field-specific list to create summaries. A criterion of word frequency (like Luhn's) was also used, but rather than combining it with another criterion to calculate a score like Edmundson, they used it to adjust the results from the keyword approach. A keyword which is frequently encountered in the *corpus* is considered less indicative of a candidate sentence for the summary than a rarely encountered keyword.

[9] At its Denver annual meeting, the president of the American Association for Information Science (ASIS) gave the 1971 ASIS award to the Ohio Sate graduate students J. Rush, R. Salvador and A. Zamora for their paper "Automatic Abstracting and Indexing" [RUS 71]

[RUS 71, POL 75] used an algorithm for sentence compression by *deleting certain words, expressions or clauses which might detract from the neatness or conciseness of a selected sentence*, which was remarkable for the time. Following their extraction, retained sentences were compressed to further reduce the length of the summary. To do this, they compiled a list of pairs of word types (when the type was adjective, verb, noun, etc.).

With the hypothesis that commas indicate the boundaries of clauses, they used grammar to classify commas. Clauses are classified according to the type of words around the commas. Introductory clauses, those ending in *that* or starting with *in* [*conclusion* | *fact* | ...], as well as appositions [10] were eliminated.

To meet the standards required by the Chemical Abstracts Service (CAS) [11], a normalization of the lexicon was required. Orthographic variants of words, abbreviations and names of chemical molecules were standardized with a bank of transformation rules. Research undertaken at the CAS into the production of summaries from scientific articles in the domain of chemistry validated the relevance of this approach. ADAM generated automatic summaries with the following characteristics:

– a length of between 10 and 20% of the original text size;

– use of the document's original terminology;

– extracts were displayed without formulas or equations;

– preliminary comments, negative results, explanations, examples and opinions were excluded from the summary;

– the objectives, results and conclusions were retained.

ADAM was the first automatic text summarization system to be commercialized.

[10] Apposition is when a noun supplements another. For example: "Computational linguistics, a booming science, has considerably helped our works". Appositions are commonly placed between commas.

[11] http://www.cas.org.

3.2. Machine learning approaches

Parameters such as the position of sentences or the presence of certain keywords have been used in the previous section to determine the importance of sentences. The next question is how can the contribution of these parameters in the selection of important sentences be determined? The answer is not simple and depends to a great extent on the type of documents to be summarized. For instance, the role of the sentence position parameter has been determined for journalistic articles: the first sentences are often the most important [BRA 95, HOV 99, LIN 97]. In scientific articles, sentences in the "conclusion" section are prioritized.

From this empirical perspective, machine learning approaches have proved to be useful because the value of parameters can be estimated from their occurrences in a previously tagged *corpus*. Countless studies have analyzed *corpora* composed of pairs {document–manually generated associated summary} in order to automatically learn the rules for generating summaries.

A common example of the use of machine learning is the calculation of weights based on the word frequency. tf.idf weighting [SPÄ 72] can be used in automatic summarization to weight words and use them during the selection stage [SAK 01, SPÄ 95]. Several variants of tf.idf are available in the literature. Appendix A.2.1 provides a detailed example of a tf.idf calculation.

3.2.1. *Machine learning parameters*

[KUP 95] describes a machine learning method inspired by Edmundson's method. A Bayes classifier, trained on a set of pairs {article–summary}, calculates the probability that each sentence will be included in the summary.

Let s be a sentence, Sum the set of sentences belonging to the summary and the parameters (features) F_i, $i = 1, 2, \ldots, k$:

$$p(s \in \text{Sum}|F_1, F_2, \ldots, F_k) = \frac{p(F_1, F_2, \ldots, F_k|s \in \text{Sum}) \cdot p(s \in \text{Sum})}{p(F_1, F_2, \ldots, F_k)} \quad [3.3]$$

assuming statistical independence of the features:

$$p(s \in \text{Sum}|F_1, F_2, \ldots, F_k) = \frac{\Pi_{i=1}^{k} p(F_i|s \in \text{Sum}) \cdot p(s \in \text{Sum})}{\Pi_{i=1}^{k} p(F_i)} \quad [3.4]$$

$p(s \in \text{Sum})$ is a constant, and the probabilities $p(F_i|s \in \text{Sum})$ and $p(F_i)$ can be estimated from a training *corpus*.

The five features used by Kupiec *et al.* were:

– sentence length cutoff feature: if the length of the sentence exceeds a certain threshold $u_1 = 5$, length $(s) > u_1$ (true/false);

– fixed-phrase feature: sentences containing any of a list of fixed phrases[12] are more likely to be in summaries: $s \cap$ dictionary (fixed-phrases) $\neq \emptyset$ (true/false);

– paragraph feature: the position of the sentence s in a paragraph: beginning – middle – end (position(s) discrete value);

– thematic word feature: the presence of thematic terms. Ranking$(s) >$ threshold u_2 (true/false);

– uppercase word feature: the presence of words in capital letters, s $\cap$ [A-Z] (true/false).

[12] Sentences which are at the beginning of an important section, such as the “Conclusion”, or sentences containing two-word terms present in a manually generated list, for instance “In summary”, “In conclusion”, etc.

The minimum sentence length to be considered was empirically set at five words. The presence of words beginning with a capital letter is associated with proper names contained in the s.

The general architecture of Kupiec *et al.*'s system is shown in Figure 3.5.

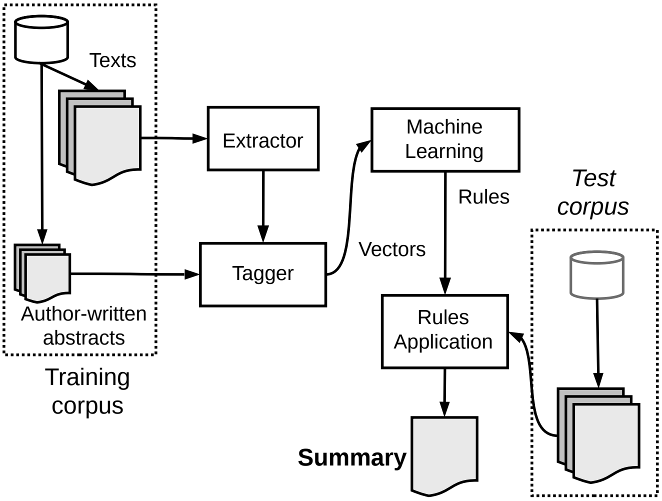


Figure 3.5. *Kupiec's [KUP 95] system based on a model which learns features*

Experiments have been carried out on a learning *corpus* composed of 188 pairs {document–summary} of scientific articles from 21 collections, with an average of 86 sentences per document. Individually, the features were ranked (from best to worst) as follows: position(s), fixed-phrases, length(s), thematic words and capital letters. The best combination of features was the following:

$$\omega(s) = \text{position}(s) + \text{fixed-phrases} + \text{lenght}(s) \qquad [3.5]$$

Although these approaches are appealing, it must be noted that, on the one hand, the learning set was very small and, on the other hand, the experiments were limited to the domain of scientific articles. This approach was built on by Aone *et al.* [AON 99] with the use of parameters such as the presence of signature-words: this method, based on the frequency of signature-words, can exploit linguistic dimensions beyond the analysis of a single word. [LIN 99] modeled the problem of extracting sentences with decision tree algorithms. Several references (baselines), such as using the position alone or the simple combination of all the parameters (addition of values), have been evaluated. The evaluation consisted of matching sentences extracted by the system with manually extracted sentences. The decision tree classifier has proved to be the best overall performing system. [LIN 99] concluded that no individual parameter nor an isolated characteristic were sufficient to generate query-guided summaries and that certain parameters were independent of others. Other learning approaches have also been tested, particularly approaches using the hidden Markov model (HMM) [CON 01] or artificial neural networks (ANN) [SVO 07]. These works enabled high performing extraction approaches to be developed, but they provided no guarantee that the summary could be exploited. Indeed, outside their original context, the selected sentences were not cohesive (for instance, the unresolved anaphoras became irritating). Moreover, the parameters used during learning may not turn out to be consistent for different types of document.

3.3. State-of-the-art approaches

As explained in section 2.3, single-document summarization systems which use extraction have a certain number of stages in common:

– *Preprocessing*: splitting into words and sentences, filtering stopwords and punctuation and word normalization;

– *Extraction*: calculating and combining the similarity measures between sentences, weighting, sorting and selecting sentences;

– *Generation*: assembly, postprocessing and probably reformulating the extracted sentences (paraphrasing).

Figure 3.6 shows the general architecture of an extraction-based single-document summarization system.

Several methods using intersentence similarity are possible: latent semantic analysis (LSA), graph-based approaches, statistical metrics, cosine, etc. This gives rise to very different algorithms for sentence weighting and sentence extraction. In the following sections, several similarity and sentence weighting methods will be studied.

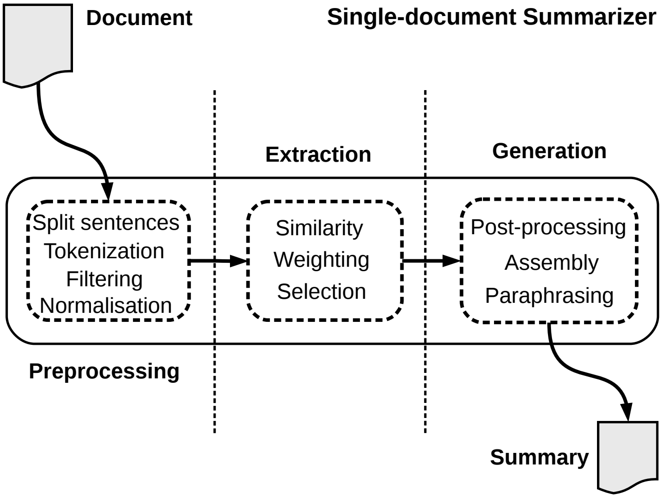


Figure 3.6. *General architecture of a extraction-based single-document summarization system*

To compare the output of several automatic summarizers, the following example will be used: Dragomir R. Radev's *corpus*, taken from the Language Engineering Summer School which took place in July 2003 at the Johns Hopkins University[13]. The *corpus* has been reproduced in Table 3.1. This *corpus* contains six sentences and 135 raw words (tokens separated by spaces). Seven systems produced the summaries shown (without the titles) in Table 3.2. The required compression rate was 33% of the length of the *corpus* in sentences.

The summaries were produced by:

– *Lead sentences baseline*;

– MEAD (see section 4.5.5);

[13] Source: http://old-site.clsp.jhu.edu/ws03/preworkshop/lecture_radev.ppt.

– WEBSUMM, summary taken directly from R. Radev's reading material;

– SWESUM, demo available at http://swesum.nada.kth.se/index-eng-adv.html;

– CORTEX (see section 3.7);

– ESSENTIAL SUMMARIZER, available at https://essential-mining.com;

– INTELLEXER SUMMARIZER, available at http://summarizer.intellexer.com/.

The values of the framework for evaluating summaries automatically (FRESA) are shown (see section 8.9.2 for this type of evaluation): the higher the score, the closer the content of the summary is to the source. It has been shown that INTELLEXER, MEAD and CORTEX produce comparable and concise summaries (CORTEX's postprocessing eliminated the text between brackets).

> ***Emergency relief by SWD.***
>
> *The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory. The people, comprising adults and children, come from 30 families. Some of them are taking temporary shelter at Lung Hang Estate Community Centre in Sha Tin, and Shek Lei Estate Community Centre and Princess Alexandra Community Centre in Tsuen Wan. The Regional Social Welfare Officer (New Territories East), Mrs Lily Wong, visited victims at Lung Hang State Community Centre this (Thursday) afternoon to offer any necessary assistance. Six victims have so far requested for Comprehensive Social Security Allowance and the applications are being processed. Social workers also escorted an 88-year old man who was feeling unwell to the Prince of Wales hospital for medical checkup.*

Table 3.1. *Corpus "Emergency relief by SWD", taken from R. Radev's reading material*

Mead; Fresa=0.23632
Some of them are taking temporary shelter at Lung Hang Estate Community Centre in Sha Tin, and Shek Lei Estate Community Centre and Princess Alexandra Community Centre in Tsuen Wan. The Regional Social Welfare Officer (New Territories East), Mrs Lily Wong, visited victims at Lung Hang State Community Centre this (Thursday) afternoon to offer any necessary assistance.

Intellexer Summarizer; Fresa=0.23632
The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory. The Regional Social Welfare Officer (New Territories East), Mrs Lily Wong, visited victims at Lung Hang State Community Centre this (Thursday) afternoon to offer any necessary assistance.

Cortex; Fresa=0.19259
The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory. The Regional Social Welfare Officer, Mrs Lily Wong, visited victims at Lung Hang State Community Centre this afternoon to offer any necessary assistance.

Essential Summarizer; Fresa=0.14789
The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory. Six victims have so far requested for Comprehensive Social Security Allowance and the applications are being processed.

Lead sentences baseline; Fresa=0.11096
The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory. The people, comprising adults and children, come from 30 families.

Websumm; Fresa=0.08381
Some of them are taking temporary shelter at Lung Hang Estate Community Centre in Sha Tin, and Shek Lei Estate Community Centre and Princess Alexandra Community Centre in Tsuen Wan.

Swesum; Fresa=0.08233
Emergency relief by SWD. The Social Welfare Department has provided relief articles and hot meals to 114 people who were affected by the rainstorm or mudslip throughout the territory.

Table 3.2. *Summaries of the document in Table 3.1 produced by the systems:* Mead, Intellexer Summarizer, Cortex, Essential Summarizer, *Lead sentences baseline,* Websumm *and* Swesum

3.4. Latent semantic analysis

LSA [14] [DEE 90, LAN 98] is a computational model which enables semantics to be represented from the following ideas: two words are semantically close if they appear in similar contexts and two contexts are similar if they contain semantically close words. The words in a (large) *corpus* are represented in the occurrence matrix S. This matrix stores, for each word in the *corpus*, the contexts in which the words appear as well as their appearance frequency. [15]

An LSA system is realized from the decomposition of S in singular values (SVD matrix). The number of dimensions in this diagonal matrix is generally set between 100 and 300.

Each word in the *corpus* is represented by a vector. The semantic proximity between two words can be measured by estimating the cosine of the angle formed by their vectors. LSA requires the use, during input, of large learning *corpora* to estimate its parameters.

3.4.1. *Singular value decomposition (SVD)*

Let $S_{[P \times N]}$ be the sentence-term matrix obtained by a vectorization process (see section 2.1.1). The matrix S can be decomposed using:

$$S = U \Sigma W^T \qquad [3.6]$$

where $U_{[\rho \times \rho]}$ is a unitary matrix, $\Sigma_{[\rho \times N]}$ is a matrix whose diagonal elements $\Sigma_{i,i} \geq 0$ and $\Sigma_{i,j} = 0$, and W^T is the transpose of the unitary matrix $W_{[N \times N]}$. The diagonal elements Σ are singular values, sorted

[14] "Latent semantic analysis (LSA) is a technique in natural language processing, in particular in vectorial semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text" (source: Wikipedia: http://en.wikipedia.org/wiki/Latent_semantic_analysis).

[15] See Chapter 18 of [MAN 08], http://nlp.stanford.edu/IR-book/pdf/18lsi.pdf, for a discussion on "Matrix decompositions and latent semantic indexing".

according to their value: $\Sigma_{[i,i]} > \Sigma_{[i+1,i+1]}$, where $\Sigma_{[1,1]}$ is the largest. S can be approached by the first k singular values:

$$S \approx U\Sigma_k W^T = U_k \Sigma_k W_k^T \qquad [3.7]$$

where U_k and W_k correspond to the first k columns of U and W. Equation [3.7] corresponds to the best approximation of rank k of S.

3.4.2. *Sentence weighting by SVD*

LSA was used in several works [DEE 90, LAN 98, FOL 98] as well as for the generation of text summaries [GON 01, LEM 05]. Gong's [GON 01] summarization algorithm is shown in Figure 3.7.

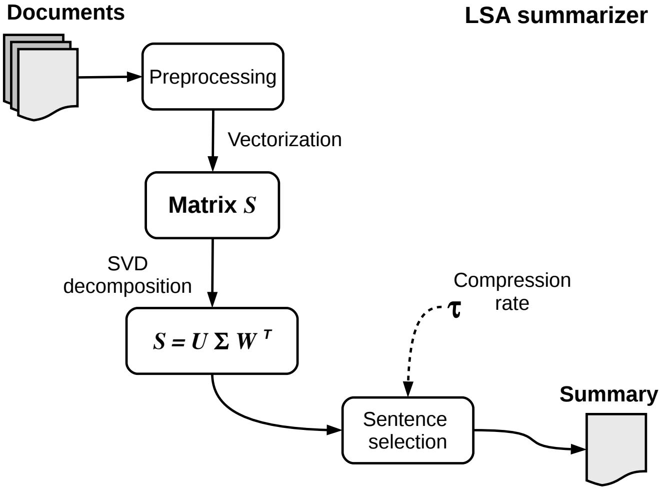


Figure 3.7. *General architecture of Gong's [GON 01] automatic summarization system based on LSA*

Sentence weighting involves the following stages:

1) Perform the SVD on S to obtain the singular value matrix Σ, and the right singular vector matrix W^T. In the singular vector space, each sentence i is represented by the column vector $i = [v_{i1}, v_{i2}, ..., v_{ir}]^T \in W^T$.

2) Select the kth right singular vector from matrix W^T.

3) Select the sentence that has the largest index value with the kth right singular vector, and include it in the summary.

4) If k reaches the compression rate τ, terminate the operation; otherwise, increment k by one and go to step 2.

The algorithm was evaluated using a set of news from *CNN Worldview* for a period of 2 months. Three human judges manually constructed summaries from 243 documents. The evaluation measured the precision P and the recall R of sentences chosen by the judges and by the system (see section 8.8.1). The performance of the LSA summarization system is comparable to the performance of a state-of-the-art summmarizer measuring relevance by similarity (similarity between each vector sentence and the document evaluated as an inner product): $P_{\text{LSA}} = 0.53, R_{\text{LSA}} = 0.61, F_{\text{LSA}} = 0.57$, $P_{\text{similarity}} = 0.52, R_{\text{similarity}} = 0.59, F_{\text{similarity}} = 0.55$, evaluated by the majority vote of three judges.

LSA has also been used to simulate cognitive processes [FOL 98]. An interesting process is the hierarchization of the importance of sentences in a text. Different models based on several cognitive hypotheses have been tested by Lemaire *et al.* [LEM 05]. One hypothesis stipulates that the more important the sentence, the more semantically close it is to the source text as a whole. LSA produces semantic proximities between a document and its sentences. This generates sentence weighting which serves to produce a summary. LSA has been used in an evaluation which involved a large number of references in French, in which the summaries of several systems were compared to summaries produced by people (abstracts and extracts) [FER 08b]. In this evaluation, three huge *corpora* were used and projected in a 300-dimensional space: LSA_le-monde, generic *corpus* of 5 million words, containing the set of articles for 1999 from the daily French newspaper *Le Monde*; LSA_children, a *corpus* intending to represent the semantic memory of children aged between 9 and 11 [LEM 05] (containing 3.3 million words from publications for children: stories, school textbooks, encyclopedias); LSA_adults, a *corpus* of 13 million words which combines a large part of the two previous *corpora* in addition to novels for adults. It is intended to represent the semantic memory of an adult. LSA is at the forefront of these methods.

LSA is also part of multilingual systems which produce news: the European Union's EMM NEWSEXPLORER generates daily news summaries; and the newswire aggregator EMM NEWSBRIEF (see section 5.6).

3.5. Graph-based approaches

Several recent studies have been conducted with regard to graph models to represent the content of textual information. Different graph applications have therefore been created: semantic disambiguation, parts-of-speech (POS) tagging, information extraction, opinion analysis and automatic text summarization. In graph-based approaches, the vertices (or nodes) are assimilated to the semantic or cognitive units (words, sentences or documents) and the edges are assimilated to the relations between these units.

With regard to automatically summarizing a text document, the most widespread approach consists of sentence weighting according to their content. The majority of automatic text summarization methods evaluated during the DUC/TAC campaigns[16] use a bag-of-words approach to represent sentences (see section 2.1 and Appendix A.2). The text is assimilated to the matrix $S_{[\rho \times N]}$ of ρ sentences and N words, which codes the occurrences of words in sentences without taking their position into account. The matrices S considered for summarization by extraction are mostly sparse binary matrices, which can be favorably represented in undirected graphs.

Graph-based algorithms have proved their effectiveness in several automatic text summarization tasks [ERK 04, MIH 04a, MIH 04b, RAD 04, MIH 05a, BAN 10]. They are also used for summarizing e-mail conversation threads [CAR 07] (see section 6.6.4), in

[16] See section 8.6, the DUC/TAC evaluation campaigns.

multisentence compression algorithms [FIL 10, BOU 13] (see section 7.6) or for tweet summarization [LIU 12] (see section 6.6.3); creating abstracts using conceptual graphs [MIR 13] (see section 7.5), keyword detection [MIH 04a] and many more.

3.5.1. PAGERANK *and SNA algorithms*

In addition to adapting large sparse matrices to a computer representation, this graph-based representation is interesting as it suggests using a large well-known family of calculations: centrality [17] of a vertex in a graph from Social Network Analysis (SNA) [18].

The PAGERANK [BRI 98] algorithm was the key to the success of online retrieval: web pages are ranked by the analysis of their *popularity* in the network, rather than by their content. This type of algorithm calculates the importance of the vertex of the graph, based on the overall information gathered from a recursive analysis of the complete graph, rather than a local analysis of a vertex.

A page i is modeled by a directed graph and its PAGERANK (popularity) score $p_R(i)$ is given by equation [3.8]:

$$p_R(i) = \frac{1-d}{N} + d \sum_{j=1}^{N} \frac{p_R(j)}{C(j)} \qquad [3.8]$$

where N is the number of vertices in the graph, $p_R(j)$ are the PAGERANK values of page j linked to i, $C(j)$ is the total number of links going out of page j, and the damping factor d which can be between [0,1] (normally set at 0.85) [19].

[17] Source: http://en.wikipedia.org/wiki/Centrality.

[18] Source: http://en.wikipedia.org/wiki/Social_network_analysis.

[19] See: http://en.wikipedia.org/wiki/PageRank concerning the confusion with formula [3.8] and whether or not the N value should be used.

The web search engine Google uses PAGERANK [PAG 98] to calculate the importance of web pages linked via hyperlinks. Intuitively, a page will have a high PAGERANK score if several pages link to it or if a few, high scoring, pages link to it. PAGERANK models a surfer who, starting from a randomly selected web page, randomly clicks on the site's links. The surfer will eventually exit the site and start all over again on another page. PAGERANK corresponds to a variant of the algorithm Power Iteration[20], which calculates the principal eigenvector of the matrix that corresponds to the graph of links between pages. The score vector therefore corresponds to the principal eigenvector of the squared matrix of links between pages [PAG 98].

Different variants of PAGERANK exist according to the definition of the valuation function and the initialization of the method. However, the principle remains the approximative calculation of the principal eigenvector of a sparse matrix in a corresponding graph.

3.5.2. *Graphs and automatic text summarization*

LEXRANK and TEXTRANK, two of the most widely used algorithms for automatic summarization by extraction which are based on the route of the graph, were not inspired by SNA but by the PAGERANK algorithm [PAG 98]. LEXRANK [ERK 04] and TEXTRANK algorithms [MIH 04a] apply the concept of graphs to sentence weighting. These methods seek to identify the most *prestigious* sentences in a graph. A document is therefore represented by a graph of text units (the ρ sentences), which are interconnected via relations from similarity calculations. The sentences are then extracted according to the criteria of centrality (or prestige) in the graph and then assembled to generate a summary. The results (the summary) show that graph-based approaches perform extremely well.

The summarization paradigm by graph algorithm is based on two stages:

[20] Source: http://en.wikipedia.org/wiki/Power_iteration.

1) constructing the graph: the sentences in a document are the vertices and the similarity between the sentences are the weighted edges. This is a text cohesion model which uses intersentence similarity;

2) implementing link analysis algorithms: this enables the n sentences with the highest ranking to be extracted. The sentences are therefore *recommended* by other sentences with similar content.

3.5.3. *Constructing the graph*

To develop algorithms for sentence weighting expressed in natural language, it is necessary to construct a graph which represents the document and interconnects the vertices with meaning relations. In automatic summarization, the most simple choice, at the vertex level, is the complete sentence. Indeed, this means that the grammaticality of the assembled segments need not be worried about while the summary is being generated.

A connection between two sentences is defined as a measure of morphological similarity. This type of relation can be seen as a "recommendation". A sentence which deals with certain concepts in the text gives the reader a recommendation with regard to the other sentences which have similar content.

For each set of documents, an undirected and valuated graph $G = (V, E)$ is constructed. G is a graph with a set of vertices S and edges E, where $E \subseteq V \times V$. For a given vertex V_i, $s_j \in In(s_i)$ represents the set of vertices j which have an edge in the direction of s_i and $\sum_{s_k \in \text{Out}(s_j)}$ represents the set of vertices connected from the vertex i by an outgoing edge. A graph G can be of the following types:

– undirected: no direction is established between the sentences. A sentence can recommend the sentences which precede or succeed it in the text;

– directed forward: a sentence only recommends the sentences which succeed it in the text. This type of graph seems to be suitable for film, fiction criticism, etc.;

– directed backward: a sentence recommends the sentences which precede it in the text. This type of graph is more suitable for journalistic articles.

The value of each edge between two vertices is evaluated with a similarity measure. Similarity can be calculated using cosine and word overlap, among others.

3.5.4. *Sentence weighting*

The graph-based approach enabled summarization algorithms which do not require a phase of machine learning features to be constructed. Sentence weighting methods use the overall information in the graph to calculate the importance of each sentence. Extraction methods implement the selection of a set of relevant sentences. This process corresponds to an identification of the most central sentences, i.e. those which contain the essential ideas in the document.

3.5.4.1. LexRank

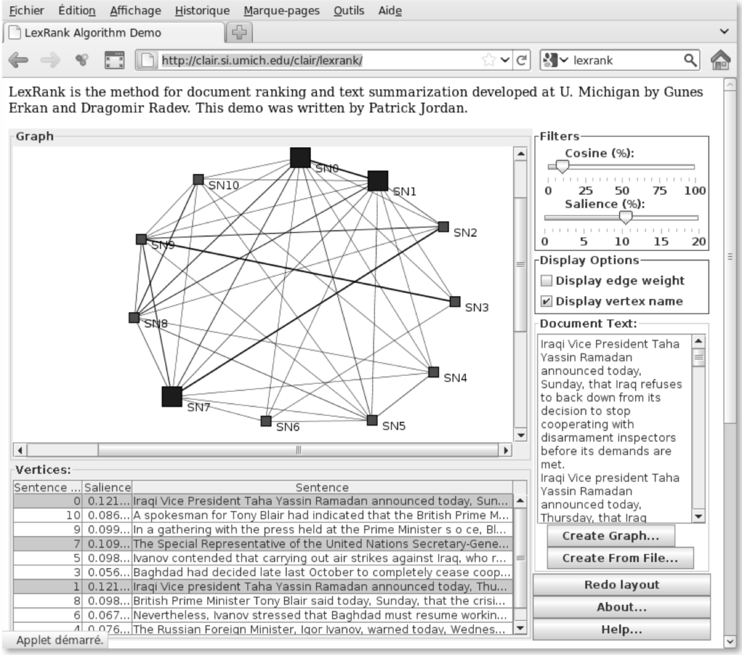
Erkan and Radev [ERK 04] developed the LexRank method, a modification of PageRank, to weight sentences. Unlike PageRank, the graph of sentences is constructed from symmetrical similarities (modified cosine measure). It is, consequently, an undirected graph. The score $\omega(\bullet)$ of each vertex s is calculated iteratively[21] by the following equation [3.9]:

$$\omega(s_i) = \frac{d}{N} + (1-d) \sum_{s_j \in In(s_i)} \frac{\text{sim}(s_i, s_j)}{\sum_{s_k \in \text{Out}(s_j)} \text{sim}(s_k, s_j)} \omega(s_j) \qquad [3.9]$$

where sim$(\bullet)$ is the cosine weighted by the inverse document frequency (idf) (see equation [A.8]), the damping factor $d = 0.85$ and N is the number of vertices.

[21] Iterations are stopped when the values of the vertices have not been modified by more than $\epsilon = 0.0001$.

LEXRANK has been used in multi-document summarization, as a component of the MEAD algorithm. A simulator of the LEXRANK algorithm can be found at http://clair.si.umich.edu/clair/demos/lexrank/. Figure 3.8 shows the interface of this demo.



LexRank is the method for document ranking and text summarization developed at U. Michigan by Gunes Erkan and Dragomir Radev. This demo was written by Patrick Jordan.

Filters

Cosine (%): 0 25 50 75 100

Salience (%): 0 5 10 15 20

Display Options

☐ Display edge weight

☑ Display vertex name

Document Text:

Iraqi Vice President Taha Yassin Ramadan announced today, Sunday, that Iraq refuses to back down from its decision to stop cooperating with disarmament inspectors before its demands are met.
Iraqi Vice president Taha Yassin Ramadan announced today, Thursday, that Iraq

Create Graph...

Create From File...

Redo layout

About...

Help...

Vertices:

Sentence ...	Salience	Sentence
0	0.121...	Iraqi Vice President Taha Yassin Ramadan announced today, Sun...
10	0.086...	A spokesman for Tony Blair had indicated that the British Prime M...
9	0.099...	In a gathering with the press held at the Prime Minister s o ce, Bl...
7	0.109...	The Special Representative of the United Nations Secretary-Gene...
5	0.098...	Ivanov contended that carrying out air strikes against Iraq, who r...
3	0.056...	Baghdad had decided late last October to completely cease coop...
1	0.121...	Iraqi Vice president Taha Yassin Ramadan announced today, Thu...
8	0.098...	British Prime Minister Tony Blair said today, Sunday, that the crisi...
6	0.067...	Nevertheless, Ivanov stressed that Baghdad must resume workin...
4	0.076...	The Russian Foreign Minister, Igor Ivanov, warned today, Wednes...

Figure 3.8. *Simulator of the graph-based system* LEXRANK. *The graph corresponds to document d1003t of the DUC'04 corpus (Agence France Presse (AFP) Arabic Newswire, 1998) [ERK 04]*

3.5.4.2. TEXTRANK

In this algorithm, the similarity values of the edges are used to weight the vertices [MIH 04a, MIH 04b]. The score of each sentence s_i is calculated iteratively until convergence by:

$$\omega(s_i) = (1-d) + d \sum_{s_j \in In(s_i)} \frac{w_{j,i}}{\sum_{s_k \in \text{Out}(s_j)} w_{jk}} \omega(s_j) \qquad [3.10]$$

where $d = 0.85$.

The weight $w_{j,i}$ between two sentences i and j is defined by:

$$w_{j,i} = \frac{\sum_{w \in i,j} C_w^i + C_w^j}{\log |s_i| + \log |s_j|} \qquad [3.11]$$

where $C_w^\bullet$ represents the occurrences of the word w in the sentence $\bullet$. Normalization by the logarithms $\log | \bullet |$ of the length of the sentences avoids long sentences being favored.

TEXTRANK has been used in single-document summarization with very good results [MIH 04b]. An example of a graph is shown in Figure 3.9. It shows the graph of sentences from the text *Hurricane Gilbert* (see Table 3.3). The summary generated by TEXTRANK with the heaviest sentences is shown in Table 3.4.

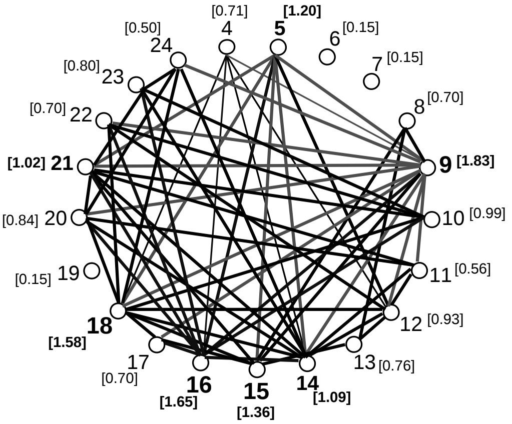


Figure 3.9. *Example of a graph for the document "Hurricane Gilbert" [MIH 04b]. The heaviest sentences are 9, 16, 18, 15, 5, 14 and 21*

It is important to emphasize that methods for sentence weighting depend entirely on the correct construction of the graph. Given that this graph is generated from intersentence similarity measures, the impact of the choice of these methods is not insignificant. [ERK 04] uses a bag-of-words to represent the sentences as a vector of N dimensions. The similarity between sentences is obtained by calculating the weighted cosine by idf between their vectorial representation. [MIH 04b] uses word overlap between sentences.

The weak point of all these similarity measures, which use words (cosine, overlap, the longest common subsequence, etc.), is that they depend on the lexicon of the documents. From the perspective of language independence, text preprocessing should use few linguistic resources. Unfortunately, in this configuration, the performance of similarity based on words falls, as words which though morphologically close are not able to be compared. A solution to this problem is the combination with similarity measures based on chains of characters [BOU 08c]. This measure enables relations between two sentences to be created, sentences which though they do not share a single word contain a number of morphologically close words. [BOU 09b] constructed graphs with mixed measures (words and characters), which enabled the performance of TEXTRANK and LEXRANK on French documents to be improved.

3.6. DIVTEX: a summarizer based on the divergence of probability distribution

Kullback–Leibler divergence $\mathcal{D}_{\mathcal{KL}}$ (or relative entropy) [KUL 51] calculates the divergence between two probability distributions. For two discrete probability distributions P and Q, the divergence $\mathcal{D}_{\mathcal{KL}}$ of Q in relation to P is:

$$\mathcal{D}_{\mathcal{KL}}(P||Q) = \frac{1}{2} \sum_{w \in P} P_w \log_2 \frac{P_w}{Q_w} \qquad [3.12]$$

For the purpose of this book, P_w is the probability distribution of words w in the document D_P and Q_w is the probability distribution of words w in the document D_Q, where $D_Q \subset D_P$.

The following specification will be used for the probability distribution of words w:

$$P_w = C_w^P/|P|; \qquad Q_w = \begin{cases} C_w^Q/|Q| & \text{if } w \in Q \\ \delta & \text{elsewhere} \end{cases} \qquad [3.13]$$

where $C_w^{P|Q}$ is the number of occurrences of the word w in the distribution P or Q; $|P|$ is the length of the document D_P in words, $|Q|$ is the length of the document D_Q in words and $\delta > 0$ is a smoothing factor.

Jensen–Shannon divergence ($\mathcal{D}_{\mathcal{JS}}$) is the symmetrized version of the divergence $\mathcal{D}_{\mathcal{KL}}$. It is defined by:

$$\begin{aligned}\mathcal{D}_{\mathcal{JS}}(P||Q) &= \frac{1}{2}\mathcal{D}_{\mathcal{KL}}(P||M) + \frac{1}{2}\mathcal{D}_{\mathcal{KL}}(Q||M); \quad M = \frac{1}{2}(P+Q) \\ &= \frac{1}{2}\sum_w P_w \log_2 \frac{2P_w}{P_w + Q_w} + Q_w \log_2 \frac{2Q_w}{P_w + Q_w} \qquad [3.14]\end{aligned}$$

with the same specification as for [3.13].

The normalized divergence values can be used to weight the sentences in a document. The idea behind Text Divergence Summarizer (DIVTEX) is as follows: for all the ρ sentences $s_\mu \in$ document D, calculate the divergence ($\mathcal{KL}$ or $\mathcal{JS}$) between the sentence μ and the rest of $\rho - 1$ sentences in the complement document $D \backslash s_\mu$, and then rank them according to divergence.

After preprocessing the corpus D, the weight of the sentence μ is given by:

$$\omega(s_\mu) = 1 - |\mathcal{D}_{\mathcal{JS}|\mathcal{KL}}[s_\mu||(D\backslash s_\mu)]|\ ; \mu = 1, \ldots, \rho \qquad [3.15]$$

Following this ranking, a summary is generated with sentences whose divergence is minimal in relation to D. However, there is no guarantee that the summary obtained in this way has a minimal divergence in relation to the complete source.

The notion of divergence has been exploited in another task concerning automatic summarization: the evaluation of summaries. In section 8.9.1, automatic methods for evaluating summaries, based on the divergence of probability distribution, are presented.

3.7. CORTEX[22]

In the vectorial approach, texts are processed as a whole (rather than separately) and are converted into a numerical representation, which differs a great deal from structural linguistic analysis, but which has been proved to have many advantages [MAN 99b]. CORTEX [TOR 02, TOR 09] is a multilingual single-document summarization system based on the vector space model (VSM) [SAL 83]. It was designed based on the philosophy of information retrieval and does not require machine learning. CORTEX combines several statistical processing operations (calculating entropy, frequential weight of segments and words, Hamming measures, etc.) with a decision algorithm. The CORTEX system consists of four stages:

1) statistical language identification, preprocessing and vectorization;

2) calculating the metrics;

3) a decision algorithm (DA) combining the information of the metrics;

4) generating the summary with a simple postprocessing.

The general architecture of CORTEX is shown in Figure 3.10. CORTEX starts by identifying the language (French, English, Spanish, etc.) of the source document. This is implemented by calculating the n-grams of letters. A standard preprocessing (filtering, splitting and normalization, see Appendix A.1) is applied to the texts and the title is detected via simple heuristics. So, a matrix S of ρ rows (sentences) × N columns (terms) is generated.

The dimension of the matrix S is the total number of word types, but in CORTEX, only the N terms with a frequency of $tf > 1$ are used [TOR 02, TOR 09]. The binary matrix $\xi_{[\rho \times N]}$, derived from the sentence-term matrix S (see section 2.1.1), is defined as follows:

$$\xi_{\mu,j} = \begin{cases} 1 & \text{if } s_{\mu,j} > 0 \\ 0 & \text{otherwise} \end{cases} \qquad [3.16]$$

[22] In Spanish, **Cortex** is *Otro Resumidor de TEXtos.*

where each component shows the presence $(\xi_{\mu,j} = 1)$ or the absence $(\xi_{\mu,j} = 0)$ of the word j in the sentence μ.

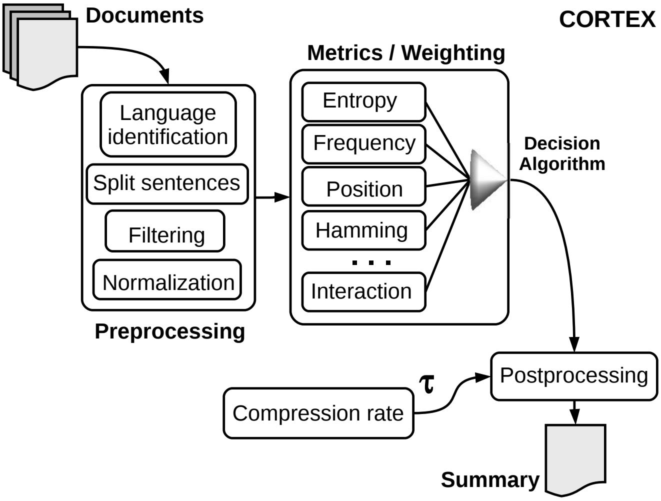


Figure 3.10. *Architecture of the automatic text summarization system* CORTEX

A set of metrics is calculated from matrices S [2.1] and ξ [3.16]. They measure the quantity of information contained in each sentence μ: the higher the metric values of a sentence s_μ, the higher its weighting.

3.7.1. *Frequential measures*

– The sum of the frequency of the word j in the sentence $\vec{s}_\mu$:

$$\omega_F(\vec{s}_\mu) = \sum_{j=1}^{N} s_{\mu,j} \qquad [3.17]$$

– The interaction of the sentences. It counts the interactions of a word j in one or several sentences:

$$\omega_I(\vec{s}_\mu) = \sum_{j=1}^{N} \sum_{\substack{k=1 \\ k\neq\mu}}^{\rho} \xi_{k,j} \qquad [3.18]$$

– Overlap: words which are contained in both the sentence s_μ and the title T of the document:

$$\omega_r(s_\mu) = 2\frac{|T \cap s_\mu|}{|T| + |s_\mu|} \qquad [3.19]$$

– Frequential sum of probability Δ: let p_j be the probability of the appearance of the word j in the text:

$$p_j = \frac{1}{t}\sum_{\mu=1}^{\rho} s_{\mu,j}\,; \quad t = \sum_{\mu=1}^{\rho}\sum_{j=1}^{N} s_{\mu,j} \qquad [3.20]$$

where t corresponds to the size of the frequential lexicon. The frequential sum of probability is calculated with:

$$\omega_\Delta(\vec{s}_\mu) = \sum_{j=1}^{N} p_j s_{\mu,j} \qquad [3.21]$$

– Sentence entropy:

$$\omega_E(\vec{s}_\mu) = -\sum_{j=1}^{N} x_j \log_2 x_j \;; x_j = \frac{s_{\mu,j}}{\sum_{j=1}^{N} s_{\mu,j}} \qquad [3.22]$$

3.7.2. *Hamming measures*

– In the Hamming matrix $H_{[N\times N]}$, each case $H_{i,j}$ exclusively represents the number of sentences using one of the terms i or j:

$$H_{i,i+1} = \left\{ \begin{matrix} 1 \text{ if } \xi_{\mu,i} \neq \xi_{\mu,j} \\ 0 \text{ elsewhere} \end{matrix} \right\}_{\substack{i=1,\ldots,N-1 \\ j=i+1,\ldots,N \\ \mu=1,\ldots,\rho}} \qquad [3.23]$$

The H matrix is an inferior triangular matrix where i represents the row and j represents the column, corresponding to the word index ($i > j$). The main idea is that if two important words (which could be synonyms) are in the same sentence, the sentence must be important. The importance of each pair of words directly corresponds to the value in the Hamming matrix H. This matrix could be used to avoid redundancy as it measures the use of terms used exclusively in two sentences.

– Hamming distance Ψ: this quantity measures the distance between pairs of words i and j within the space of segments. Each word is represented by a binary vector $\vec{\xi_i} = \{0,1\}^{\rho}$:

$$\omega_{\Psi}(s_{\mu}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} H_{i,j} \text{ if } (\xi_i, \xi_j) \neq 0 \qquad [3.24]$$

– Hamming weight ϕ: each sentence is weighted by a weight equal to its lexicon:

$$\omega_{\phi}(\vec{s}_{\mu}) = |\vec{s}_{\mu}| = \sum_{j=1}^{N} \xi_{\mu,j} \qquad [3.25]$$

– The sum of the Hamming weight of words Θ: the weight of the words on each column of the matrix ξ can be measured:

$$\psi_j = \sum_{\mu=1}^{\rho} \xi_{\mu,j};\, j = 1, \ldots, N \qquad [3.26]$$

The sum of the Hamming weight of words by sentence is:

$$\omega_{\Theta}(\vec{s}_{\mu}) = \sum_{j=1}^{N} \psi_j \times \xi_{\mu,j} \qquad [3.27]$$

3.7.3. *Mixed measures*

– The heavy Hamming weight is the product of the Hamming weight of the sentence ϕ multiplied by the Hamming weight of words Θ:

$$\omega_{\Pi}(\vec{s}_{\mu}) = \omega_{\phi}(\vec{s}_{\mu}) \times \omega_{\Theta}(\vec{s}_{\mu}) \qquad [3.28]$$

– Sum of the Hamming weights of words by frequency Ω. Sum of the Hamming weights of words j in each sentence μ, multiplied by the word frequency:

$$\omega_{\Omega}(\vec{s}_{\mu}) = \sum_{j=1}^{N} \omega_{\phi}(\vec{s}_{\mu}) \times s_{\mu,j} \;;\; \text{ if } s_{\mu,j} \neq 0 \qquad [3.29]$$

– Position of the sentence. The most important sentences are supposed to be at the beginning and/or the end of the document. The score of a sentence μ is modeled by a parabola of vertex (ρ MOD 2, 0)[23] or by an inverse function of the position:

$$\omega_x(\vec{s}_\mu) = \begin{cases} (\mu - (\rho \text{ MOD } 2))^2 \\ \mu^{-1} \end{cases} \quad ; \mu = 1, \ldots, \rho \qquad [3.30]$$

– Cosine of the angle between the title $\vec{T}$ and each sentence $\vec{s}_\mu$:

$$\omega_A(\vec{s}_\mu) = \cos(\vec{T}, \vec{s}_\mu) = (\vec{T} \times \vec{s}_\mu)/\|\vec{T}\|\|\vec{s_\mu}\| \qquad [3.31]$$

3.7.4. *Decision algorithm*

The decision algorithm (DA) retrieves coded information by metrics and weights sentences according to a voting strategy. The idea is, given a set of v independent voters $\omega_v(s)$, to find an optimal decision. To calculate the score of the sentence s, the DA combines the normalized output $\|\omega_v(s)\|$ between [0,1] of the metrics. Two averages are calculated: *positive* tendency when $\omega_v(s) > \frac{1}{2}$ and *negative* tendency when $\omega_v(s) < \frac{1}{2}$[24]:

$$\alpha(s) = \sum_{\substack{v \\ \|\omega_v(s)\| > \frac{1}{2}}} \left(\|\omega_v(s)\| - \frac{1}{2} \right); \ \beta(s) = \sum_{\substack{v \\ \|\omega_v(s)\| < \frac{1}{2}}} \left(\frac{1}{2} - \|\omega_v(s)\| \right) \qquad [3.32]$$

where v is the index of the metric, $\alpha(s)$ is the sum of positive tendency metrics (those that give a high score) and$\beta(s)$ is the sum of negative tendency metrics (those that give a low score).

[23] The module (MOD) of a quantity x divided by y represents the remainder of the division x/y.

[24] The simple average can be ambiguous if the value is $\frac{1}{2}$; the decision algorithm therefore eliminates these sentences.

The weight of a sentence s is therefore calculated (each average is divided by the total number of metrics v) as:

$$\omega(s) = \frac{1}{2} \begin{cases} +\frac{\alpha(s)}{v} & \text{if } \alpha(s) > \beta(s) \\ -\frac{\beta(s)}{v} & \text{elsewhere} \end{cases} \qquad [3.33]$$

CORTEX has been tested on single-document summarization tasks in English, French and Spanish [TOR 02] and on multi-document tasks: good results were obtained [BOU 07a, TOR 09]. [VIV 10a] combined CORTEX with semantic summarization systems; [CUN 07a] combined it with linguistic approaches. [SAN 07] used CORTEX in query expansion to improve the performance of information retrieval (IR) systems.

3.8. ARTEX: another summarizer based on the vectorial model

ARTEX[25] is another simple extractive algorithm. The main idea is: first, the text is represented in a suitable space model (VSM). Then, an average document vector that represents the average (the "global topic") of all sentences vectors is constructed. At the same time, the "lexical weight" for each sentence, i.e. the number of words in the sentence, is obtained. After that, the angle between the average document and each sentence is calculated. Narrow angles α indicate that the sentences near the "global topic" should be important and are therefore extracted. See Figure 3.11 for the VSM of words: ρ vector sentences and the average "global topic" are represented in an N-dimensional space of words.

Next, a weight for each sentence is calculated using their proximity with the "global topic" and their "lexical weight". In Figure 3.12, the "lexical weight" is represented in a VSM of ρ sentences. Narrow angles indicate that words closest to the "lexical weight" should be important.

[25] In French, ARTEX is *Autre Résumeur de TEXtes*.

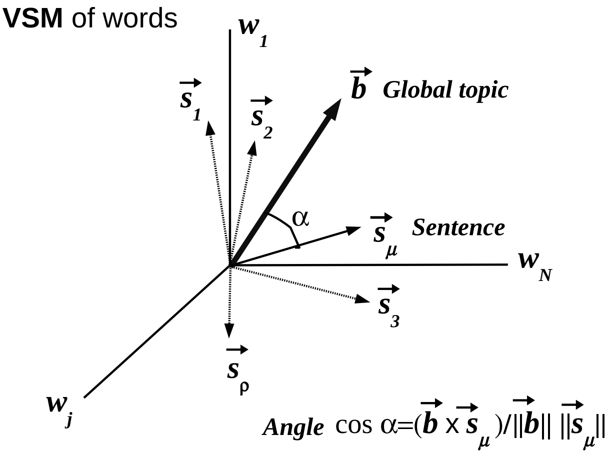


Figure 3.11. *The "global topic" in a vector space model of N words*

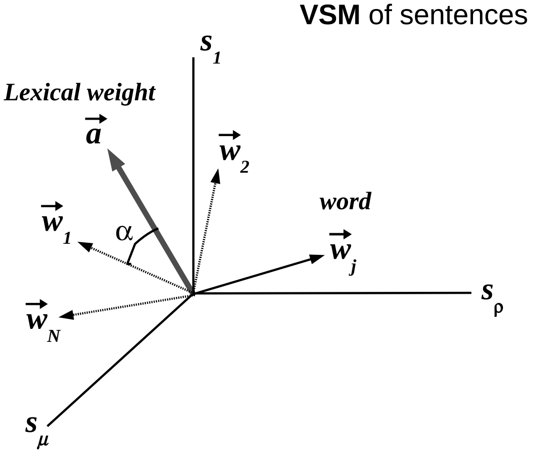


Figure 3.12. *The "lexical weight" in a vector space model of* ρ *sentences*

Finally, the summary is generated concatenating the sentences with the highest scores following their order in the original document. Formally, ARTEX algorithm computes the score of each sentence by calculating the inner product between a sentence vector, an *average pseudo-sentence* vector (the "global topic") and an *average* *pseudo-word* vector (the "lexical weight"). Once the preprocessing is complete, a matrix $S_{[\rho \times N]}$ (N words and ρ sentences) is created. Let $\vec{s}_\mu = (s_{\mu,1}, s_{\mu,2}, \ldots, s_{\mu,N})$ be a vector of the sentence $\mu = 1, 2, \ldots, \rho$. The *average pseudo-word* vector $\vec{a} = [a_\mu]$ was defined as the average number of occurrences of N words used in the sentence $\vec{s}_\mu$:

$$a_\mu = \frac{1}{N} \sum_j s_{\mu,j} \qquad [3.34]$$

and the *average pseudo-sentence* vector $\vec{b} = [b_j]$ was defined as the average number of occurrences of each word j used through the ρ sentences:

$$b_j = \frac{1}{\rho} \sum_\mu s_{\mu,j} \qquad [3.35]$$

The weight of a sentence $\vec{s}_\mu$ is calculated as follows:

$$\omega(\vec{s}_\mu) = \left(\vec{s} \times \vec{b}\right) \times \vec{a} = \frac{1}{N\rho} \left(\sum_{j=1}^{N} s_{\mu,j} \times b_j \right) \times a_\mu \, ; \; \mu = 1, 2, \ldots, \rho \qquad [3.36]$$

The $\omega(\bullet)$ computed by equation [3.36] must be normalized between the interval [0,1]. The calculation of $(\vec{s} \times \vec{b})$ indicates the proximity between the sentence $\vec{s}_\mu$ and the *average pseudo-sentence* $\vec{b}$. The product $(\vec{s} \times \vec{b}) \times \vec{a}$ weights this proximity using the *average pseudo-word* $\vec{a}$. If a sentence $\vec{s}_\mu$ is near $\vec{b}$ and their corresponding element a_μ has a high value, then $\vec{s}_\mu$ will have a high score. Moreover, when a sentence $\vec{s}_\mu$ is far from a main topic (i.e. $\vec{s}_\mu \times \vec{b}$ is near 0) or their corresponding element a_μ has a low value (i.e. a_μ is near 0), then $\vec{s}_\mu$ will have a low score.

It is not really necessary to divide the scalar product by the constant $\frac{1}{N\rho}$ because the angle $\alpha = \arccos{(\vec{b} \times \vec{s}_\mu)}/||\vec{b}||||\vec{s}_\mu||$ between $\vec{b}$ and $\vec{s}_\mu$

is the same if $\vec{b} = \vec{b'} = \sum_{\mu} s_{\mu,j}$. The element a_μ is only a scale factor that does not modify α [TOR 12]:

$$\omega(s_\mu)* = \frac{1}{\sqrt{N^5 \rho^3}} \left(\sum_{j=1}^{N} s_{\mu,j} \times b_j \right) \times a_\mu \, ; \, \mu = 1, 2, \ldots, \rho \qquad [3.37]$$

The term $1/\sqrt{N^5 \rho^3}$ is a constant value and then $\omega(\bullet)$ (equation [3.36]) and $\omega(\bullet)*$ (equation [3.37]) are both equivalent.

3.9. ENERTEX: a summarization system based on textual energy

A system based on paradigms which are very different to Natural Language Processing will now be discussed. ENERTEX is a system based on statistical physics. The algorithm models a document as a neural network from which the textual energy is deduced. To present the formalism of the ENERTEX model, a certain amount of specialized discourse about statistical physics and ANN is required, but it will stay at a level which is suitable for the reader. Some models from statistical physics have been used in NLP tasks. [TAK 05] presented an algorithm in which the spin stages represent the positive or negative polarity of the words. This enables semantic orientations to be extracted to classify the (positive or negative) opinions in a corpus.

3.9.1. *Spins and neural networks*

A magnetic system is composed of a set of N small magnets called spins, which interact with each other. These spins can move in several directions[26]. The most simple case is the Ising model, which only considers two possible directions: upward directed spins (coded as $\uparrow$, $+$ or 1) or downward directed spins (coded as $\downarrow$, $-$ or 0). The Ising model

[26] In quantum mechanics and particle physics, spin is an intrinsic form of angular momentum carried by elementary particles, composite particles (hadrons) and atomic nuclei. Spin is solely a quantum-mechanical phenomenon; it does not have a counterpart in classical mechanics (source: Wikipedia: http://en.wikipedia.org/wiki/Spin_(physics)).

has been used in a large variety of systems which can be described by binary variables [HER 91]. A system of N binary units (neurons) has $\rho = 1, \dots, 2^N$ possible configurations (or patterns) to store.

John Joseph Hopfield introduced the concept of energy in neural networks [HOP 82] from an analogy with magnetic systems [27]. In Hopfield's model, spins correspond to neurons, which are linked by synaptic weights and which interact according to Hebb's learning rule [28]. This rule develops the synaptic weight $J_{i,j}$ proportionately to the correlation between the states of neurons i and j, in all the ρ patterns [HER 91]:

$$J_{i,j} = \sum_{\mu=1}^{\rho} s_{\mu,i} s_{\mu,j} \qquad [3.38]$$

where $s_{\mu,i}$ and $s_{\mu,j}$ are the states of neurons i and j and autocorrelations are not taken into account. Therefore, necessarily $i \neq j$ on the N units of the network.

Since $J_{i,j}$ depends exclusively on the state of spins i and j, this interaction rule is local. The model is also known as an associative memory [CHA 06, KAN 87, KOS 88]. This memory has the capacity to store and retrieve a certain number of configurations from the system. The learning rule [3.38] transforms these configurations into attractors (local minimums) of the thus defined energy function [HOP 82]. These attractors are assimilated to "recalls" stored in the memory. Energy varies according to the configuration of the system, i.e. the activation states or inhibition of neurons.

[27] Readers interested in artificial neural networks and its connection to Hopfield's theory can refer the excellent book by Hertz *et al.* [HER 91]. There are several additional electronic resources available on R. Palmer's Website: http://www.phy.duke.edu/palmer/HKP/. For the applications of neural networks for NLP, consult Memmi D., Gabi K., Meunier J.-G., "Dynamical knowledge extraction from texts by ART networks", *NEURAP'98,* Marseille, France, 1998.

[28] Indeed, the literature provides several learning rules for this type of network. See, for instance, [HER 91].

If we present a pattern μ to the network, each neuron receives a local field h_i induced by the other N neurons (see Figure 3.13), calculated by equation [3.39]:

$$h_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} J_{i,j} s_j \qquad [3.39]$$

The neurons will align themselves according to their field h_i to retrieve the – closest – pattern associated with the presented pattern μ. [FER 07]'s approach only concerns the properties of the energy of the system (equation [3.40]) and not its capacity to memorize:

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} s_i \ J_{i,j} \ s_j \qquad [3.40]$$

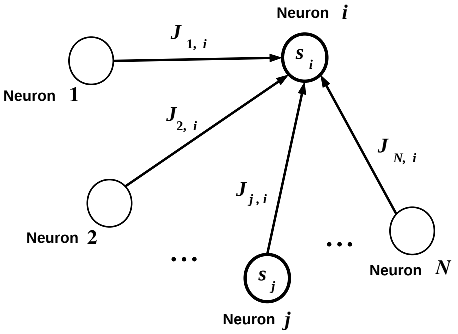


Figure 3.13. *Field h_i received by the neuron s_i, induced by the other N-1 neurons*

3.9.2. *The textual energy similarity measure*

The vectorial model transforms a document into a term-sentence matrix S. The matrix S can be seen as the set of ρ configurations of a system, where it is possible to calculate its energy. The documents are preprocessed with standard algorithms for filtering functional words [29], for normalization and lemmatization to reduce the dimensionality [MAN 99b, POR 80]. A bag-of-words representation produces a term-sentence matrix $S_{[\rho \times N]}$ of frequency/absences, composed of $\mu = 1, 2, \ldots, \rho$ sentences (patterns or rows); $\vec{s}_\mu = \{s_{\mu,1}, \ldots, s_{\mu,j}, \ldots, s_{\mu,N}\}$, and a lexicon of $j = 1, 2, \ldots, N$ terms (spins), as indicated in equation [2.1].

In this matrix, the presence of the word j represents a spin s_j with a given magnitude by its frequency tf_j (its absence by 0), and a sentence $\vec{s}_\mu$ is therefore a chain of N spins. [HOP 82]'s model differs in two ways: S is an entire matrix (its elements take the values of the occurrence of words) and the elements $J_{j,j}$ are used because this autocorrelation enables the interaction of the word_j between the ρ sentences to be established, which is important in NLP. To calculate the interactions between the N terms in the lexicon, Hebb's rule is applied in a matrix format:

$$J = S^T \times S \qquad [3.41]$$

Each element $J_{i,j} \in J_{[N \times N]}$ is equivalent to the calculation of [3.38]. Textual energy (equation [3.40]) can therefore be expressed as:

$$E = -\frac{1}{2} S \times J \times S^T \qquad [3.42]$$

An element $E_{\mu,\nu} \in E_{[\rho \times \rho]}$ represents the interaction energy between the sentences (patterns) μ and ν. The energy matrix [3.42] can be written as:

$$E = -\frac{1}{2} S \times (S^T \times S) \times S^T = -\frac{1}{2}(S \times S^T)^2 \qquad [3.43]$$

The nature of the links between sentences, which textual energy causes, has been interpreted through graph theory. The textual energy matrix E is the adjacency matrix of the graph $G(S \times S^T)^2$. From this

[29] See Appendix A.1.2.

representation, it can be deduced that this matrix links both sentences with common terms, given that it includes the intersection graph, as well as sentences with the same neighborhood which do not necessarily have the same lexicon [FER 08a, FER 09a].

3.9.3. *Summarization by extraction and textual energy*

Textual energy has been used to automatically cluster short definitions [MOL 10b]. Another application consists of using the textual energy of a sentence, seen as a spectrum, and comparing it to the spectrums of other sentences. A statistical test can indicate its degree of similarity to other sentences that are located nearer to it. This degree enables the topic boundaries of a document to be identified. These applications are out of the scope of this chapter, but interested readers can consult [FER 08a]. This work is focused on the applications of textual energy in automatic text summarization [FER 07, FER 08a, FER 09a]. The textual energy of a sentence ν has been used to weight the sentences in a document. This led to ENERTEX, a strategy for automatic summarization by extraction. The ENERTEX automatic summarization algorithm is composed of three modules (Figure 3.14). The first module conducts a standard preprocessing: filtering, normalization of words and vectorization (see Appendix A.1.2). The second module applies the spin model and carries out the calculation of the textual energy matrix (equation [3.43]).

The weight of the sentence s_ν is obtained by adding up its absolute energy values (equation [3.44]):

$$\omega(s_\nu) = \sum_{\mu} |E_{\mu,\nu}| \qquad [3.44]$$

This process is repeated for the ρ sentences in the document, which generates a sorted list. The relevant sentences are selected from those with the highest absolute energy. Finally, the third module generates summaries with a light postprocessing operation: eliminating parentheses and normalizing punctuation.

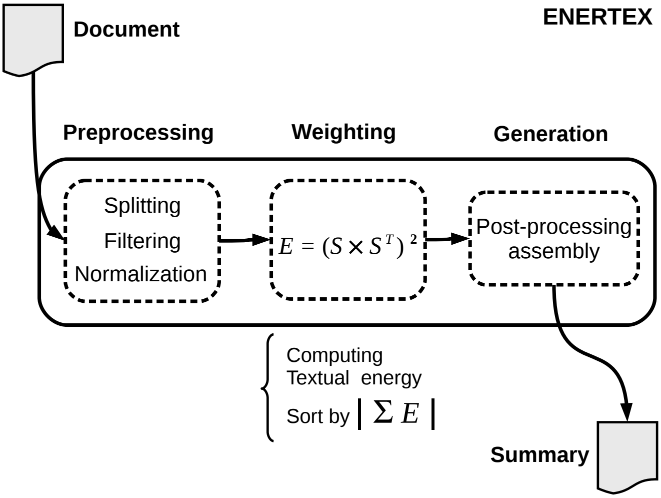


Figure 3.14. *Architecture of the automatic summarization system* ENERTEX

The performance of ENERTEX has been compared, on the basis of French and English *corpora*, to several systems[30]: MEAD, COPERNIC SUMMARIZER, *Word*, PERTINENCE SUMMARIZER, CORTEX and a random baseline. The compression rate was found to be varied (according to the length of the text) and was expressed as a percentage of the number of sentences in the source.

Figure 3.15 shows the average of the ROUGE-2 measures[31] versus the -SU4 measures for the methods used in the comparison. For reasons of clarity, only the standard deviation which corresponds to the value of -SU4 is shown. ENERTEX and CORTEX gave very similar performances (with a -SU4 moderate standard deviation) and stood apart from the other systems.

[30] More details can be found in [TOR 07].

[31] See Chapter 8 concerning the automatic evaluation of systems, in particular the ROUGE method 8.8.2.1.

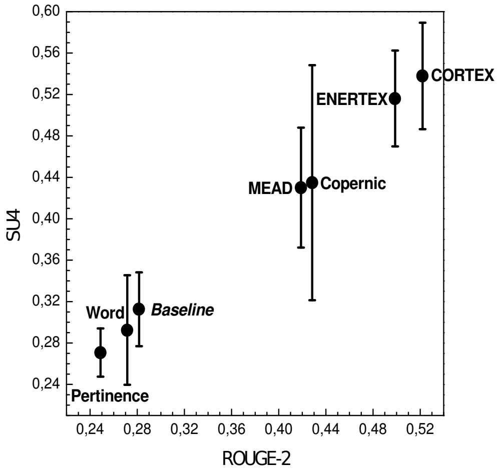


Figure 3.15. *The average* ROUGE *scores for single-document summaries generated by different systems*

Example

Table 3.3 shows the text *Hurricane Gilbert* taken from the Website http://clair.si.umich.edu/clair/CSTBank/samples/AP880911-0016.txt. It contains 24 sentences and 363 raw words. This corpus will be used to evaluate the performance of several single-document summarization systems and two human references.

The performance of five single-document summarization systems is shown in Table 3.4. The compression rate $\tau = 30\%$ of the length of the *corpus* in sentences. The reference summaries and the TEXTRANK output have been taken directly from [MIH 04b]; the summary of ESSENTIAL SUMMARIZER is from the website https://essential-mining.com. The values of the FRESA evaluation are shown[32]: the higher the score, the closer the content of the summary is to the source.

[32] See section 8.9.2 for this type of evaluation.

Hurricane Gilbert Heads Toward Dominican.

AP880911-0016 §1 AP-NR-09-11-88 0423EDT §2 r i BC-HurricaneGilbert 09-11 0339 §3 BC-Hurricane Gilbert, 0348 §4 Hurricane Gilbert Heads Toward Dominican §5 By Ruddy Gonzalez. §6 Associated Press Writer. §7 Santo Domingo, Dominican Republic (AP) §8 Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. §9 The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph. §9 "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. §10 Cabral said residents of the province of Barahona should closely follow Gilbert's movement. §11 An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo. §12 Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. §13 The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. §14 The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. §15 The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. §16 Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast. §17 There were no reports of casualties. §18 San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night. §19 On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. §20 Residents returned home, happy to find little damage from 80 mph winds and sheets of rain. §21 Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane. §23 The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month. §24

Table 3.3. *"Hurricane Gilbert" corpus, taken from http://clair.si.umich.edu/clair/CSTBank/samples/AP880911-0016.txt. Each of the 24 sentences end with §*

CORTEX*;* FRESA = 0.21665

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast.

TEXTRANK*;* FRESA = 0.21665

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast.

REFERENCE 2*;* FRESA = 0.15575

Hurricane Gilbert swept toward the Dominican Republic Sunday with sustained winds of 75 mph gusting to 92 mph. Civil Defense Director Eugenio Cabral alerted the country's heavily populated south coast and cautioned that even though there is no need for alarm, residents should closely follow Gilbert's movements. The U.S. Weather Service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday. Gilbert brought coastal flooding to Puerto Rico's south coast on Saturday. There have been no reports of casualties. Meanwhile, Hurricane Florence, the second hurricane of this storm season, was downgraded to a tropical storm.

REFERENCE 1*;* FRESA = 0.10616

Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas. Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

ESSENTIAL SUMMARIZER; FRESA = 0.10559

"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday. Cabral said residents of the province of Barahona should closely follow Gilbert's movement. On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast. Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.

Table 3.4. *Summaries of the "Hurricane Gilbert" corpus (Table 3.3) generated by Cortex,* TEXTRANK, ESSENTIAL SUMMARIZER *and two human references*

3.10. Approaches using rhetorical analysis

Within deep parsing approaches, there are approaches based on rhetorical analysis. Rhetorical structure theory (RST) makes the assumption that a text is divided into a hierarchical structure of elementary discourse units (EDUs) [MAN 87]. In this organization, the EDUs are either satellites or nuclei. A satellite needs a nucleus to be intelligible, but the opposite is not true. These units are asymmetrically linked by a rhetorical relation. The relations derive from the discursive objectives produced by the speaker. Some satellites are more centered in relation to the discursive objectives than others.

According to RST, there is a limited number of universal and fixed relations which can be easily modeled, at least theoretically: elaboration, condition, reformulation, contrast, illustration, opposition, antithesis, concession, justification, etc. The relations can be marked grammatically, usually with the help of conjunctions such as "but", "then" and "so", or they can be inferred from the context or machine learned. Diagrams are used to specify the structural composition of a text. The most frequently used diagram is two segments of text (sentences, clauses, etc.) linked so that one of them plays a specific role in relation to the other. For instance, when an affirmation of a fact is followed by an elaboration to support it, RST considers the affirmation more important (the nucleus) than its elaboration (the satellite of this nucleus). This notion of importance, called "nuclearity", is central to RST. Representations constructed in this way could be used to determine the most important segments in a text.

These ideas have been used by Marcu and Ono *et al.* [MAR 98, MAR 00b, ONO 94] in systems aimed at producing summaries. Their approaches constructed a rhetorical tree based on the presence of explicit markers in the text. This model provides an exploitable formalization of the rhetorical tree structure. Marcu [MAR 00b] created a prototype of an automatic discourse analyzer for English, based on, to a large extent, the use of discursive markers. Once the discursive structure of the text has been automatically created, an algorithm is applied which weights and orders each EDU for the tree structure of the discourse (the higher the element in the structure, the more significant its weight). The selection of sentences for the summary is realized with the heaviest elements (those with the lowest weights will be eliminated). In accordance with the desired length of the summary, more or fewer elements are chosen, but they are always in the order determined by the algorithm.

Marcu's [MAR 98, MAR 00b] approach uses clauses as EDUs. These units are promoted recursively. A clause at the n level of the tree is promoted to a higher level if it is the nucleus of the relation at level $n - 1$. Table 3.5 shows an example of discursive analysis for a text taken from the RST Website[33]. The beginning of an extract from an article in *Scientific American* (October 1972) has been segmented into numbered units. In this representation (Figure 3.16), the following units (EDU) are satellites: EDU 1, EDU 3 and the group (*SPAN*) 2-3. The nuclei are EDU 2, EDU 4, EDU 5, the group 4-5 and the group 2-5.

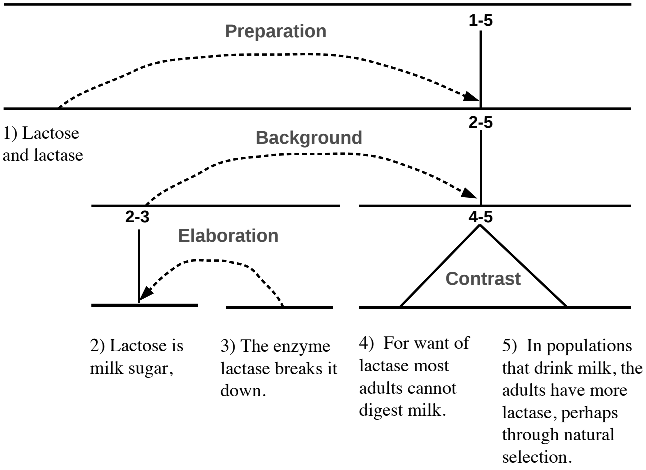

1-5
Preparation
1) Lactose and lactase
2-5
Background
2-3
Elaboration
4-5
Contrast
2) Lactose is milk sugar,
3) The enzyme lactase breaks it down.
4) For want of lactase most adults cannot digest milk.
5) In populations that drink milk, the adults have more lactase, perhaps through natural selection.

Figure 3.16. *Example of a rhetorical tree (Scientific American, October 1972, extract from http://www.sfu.ca/rst/01intro/intro.html)*

[33] http://www.sfu.ca/rst/01intro/intro.html. We thank Maite Taboada for providing this example.

A summary eliminating the satellites should maintain EDU 4 and 5:

> "For want of lactase most adults cannot digest milk. In populations that drink milk, the adults have more lactase, perhaps through natural selection."

EDU	Content
1	Lactose and lactase.
2	Lactose is milk sugar.
3	The enzyme lactase breaks it down.
4	For want of lactase most adults cannot digest milk.
5	In populations that drink milk, the adults have more lactase, perhaps through natural selection.

Table 3.5. *Example of an extract from an article in Scientific American, taken from the RST Website http://www.sfu.ca/rst/*

In the approach developed by Ono *et al.* [ONO 94], the minimum analysis unit is the sentence. Sentences are penalized according to their rhetorical role in the tree: a penalty of 0 is given to each nucleus and a penalty of 1 to the satellites.

Table 3.6 shows an example taken from [ONO 94]. This *corpus* contains six sentences, represented in the tree in Figure 3.17(a), with their rhetorical relations *Example, Extension, Special,* etc. The penalty for each sentence is shown in Figure 3.17(b).

A long extract will be generated by choosing the nodes 1, 2, 3 and 6. In an ultra-summary, only sentence 6 will be displayed. An intermediate-length summary will contain sentences 1, 2 and 6:

> In the context of discrete-time signals, zero-crossing is said to occur if successive samples have different algebraic signs. The rate at which zero crossings occur is a simple measure of the frequency content of a signal. Thus, the average zero-crossing rate gives a reasonable way to estimate the frequency of a sine wave.

Sentence	Digital Processing of Speech Signals
1	In the context of discrete-time signals, zero crossing is said to occur if successive samples have different algebraic signs.
2	The rate at which zero crossings occur is a simple measure of the frequency content of a signal.
3	This is particularly true of narrowband signals.
4	For example, a sinusoidal signal of frequency F0, sampled at a rate Fs, has Ff/Fs samples per cycle of the sine wave.
5	Each cycle has two zero crossings so that the long-term average rate of zero crossings is Z = 2 F0/Fs.
6	Thus, the average zero crossing rate gives a reasonable way to estimate the frequency of a sine wave.

Table 3.6. *Corpus "Digital Processing of Speech Signals" taken from [ONO 94]*

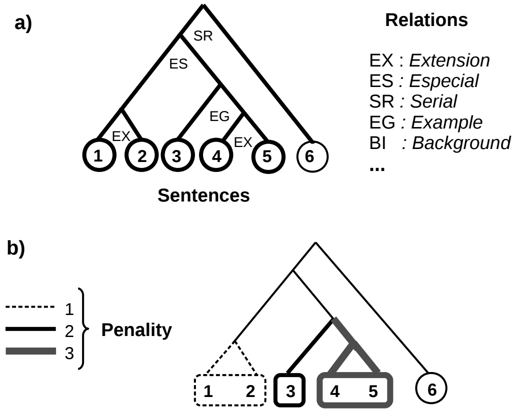


Figure 3.17. *Example of a rhetorical tree [ONO 94]*

There are other approaches which exploit the discursive structure of texts, but they approach the task from a different angle. For instance, Teufel and Moens [TEU 02] suggested a method for the summarization of scientific articles (in particular, documents dealing with computational linguistics) based on the rhetoric of the declarations they contain. They developed an algorithm which uses a hierarchical structure based on rhetoric, but which is not restricted to the seven categories defined by RST. Next, the algorithm chooses the content which must be included in the summary based on these categories.

Teufel and Moens' [TEU 02] approach differs from the traditional estimations based on RST relations [MAR 98, ONO 94] in two ways. First, their algorithm is based on the idea that relevance and the role of certain elements can be determined without analyzing the hierarchical structure of the whole text, while estimations based on RST conduct an examination of the discursive hierarchical representation of the whole text. Second, [TEU 02] seeks to capture relevant information with the rhetoric analysis of a text, while estimations based on RST give each element a status (nucleus or satellite) in relation to other elements.

RST has been used in a wide range of applications, from automatic summarization to information extraction, from text generation to automatic translation and so on [TAB 06]. Experiments conducted on the use of RST for the production of summaries are promising [CUN 08]. Nevertheless, the absence of a high performing automatic tagger independent from language, which would enable the structural composition of documents to be identified, presents a real problem. However, the task of automatic sentence compression has recently been tackled by an algorithm that combines discursive segmentation and textual energy [MOL 13c]. Indeed, segmenting a sentence into discursive units is a less difficult task than finding discursive relations. Discursive segmenters are currently available for English [MAR 00a], Portuguese [PAR 08], French [AFA 10] and Spanish [CUN 12], among others. Segmenting into EDUs then enables intersentence weighting to be carried out. This is interesting in the context of sentence compression, as some EDUs could be eliminated according to their weight (see section 7.7.3). Deep approaches to RST fall beyond the scope of this book. If interested, a reader could consult, for example, [MAN 87, MAN 88, TAB 06] and the RST website http://www.sfu.ca/rst/, which provides several sources of information, articles, examples, software and *corpora*.

3.11. Summarization by lexical chains

Lexical chains try to identify a link of cohesion between parts of a text by identifying the relevant relations between their words. A chain is identified if it contains terms referring to the same object. [BAR 97] defined an algorithm for text summarization based on the concept of lexical chains. It is composed of the following stages:

– The text is segmented with different degrees of granularity according to the application.

– To identify a candidate chain:

- the texts are preprocessed. The named entities (NEs) are recognized and annotated with POS tags;

- The candidate chain (extra-strong, strong and medium-strong relations) is created from common nouns, proper nouns, NEs, noun phrases and pronoun definition.

– The chains are marked so that the strong chains can be identified. The sentences are weighted and sorted into a list according to the chains they contain. The sentences which are crossed by a large number of chains are considered the most relevant.

The algorithm uses a set of resources: the *WordNet* thesaurus, POS-tagging, a shallow parser to identify the nominal groups and a segmentation algorithm. A certain number of sentences are extracted from this weighted list until the determined length of the summary is reached. To evaluate the algorithm for lexical chains, the authors calculated the precision and the recall for a news corpus. A set of summaries of the length $\tau = 10\%$ and 20% was also generated with lexical chains and with Microsoft's summarizer. The documents were chosen from the TREC collection. Table 3.7 shows the results of this experiment in terms of precision P and recall R (see section 8.8.1), from which it can be seen that it is not difficult to outperform Microsoft's summarizer.

3.12. Conclusion

The problems of single-document automatic summarization, a seemingly simple task, are far from being resolved. The paradigm of sentence extraction: preprocessing – weighting, selection – generation has proved to be effective for the production of summaries. Consequently, extractive approaches using a superficial analysis (like the distribution of words, the position of sentences, the co-occurrence of words, etc.), which are simple to develop and relatively adaptable to other domains or languages, are unquestionably the widely used methods. LSA and graph-based approaches (using several similarity measures) have performed very well. Among other methods (CORTEX, ARTEX, DIVSUM, etc.), the concept of textual energy, based on approaches using neural networks, has been presented. Tests on the DUC *corpora* gave very high performing results.

Compression rate	**Microsoft**		**Lexical chains**	
τ	P	R	P	R
10 %	0.33	0.37	**0.61**	**0.67**
20 %	0.32	0.39	**0.47**	**0.64**

Table 3.7. *Results for lexical chains summarizer*

On the other hand, rhetorical approaches are advantageous in that they analyze texts with a finer level of granularity than the sentence. This may produce higher quality summaries. However, it must be noted that extractive algorithms are close to reaching the limits of their performance [GEN 10]. Combining them with other paradigms (sentence compression, identifying anaphoras, reformulation, etc.) could take us to the next level and a step closer to what humans can do.

4

Guided Multi-Document Summarization

In the previous chapter, the problems, foundations and solutions relating to the generation of single-document summaries were studied. This chapter will present the special features and the proposed approaches, which are generally guided by a topic, to automatic multidocument summarization. This type of text summarization is also known as personalized or oriented summarization. We will see that the sentence extraction approaches which we studied previously are also relevant for multidocument summarization, though they need to be adapted to the particular problems of this new task. The issue of redundancy, which did not occur in single-document extraction, is omnipresent in multidocument summarization. Redundancy has a strong impact on the coherence and the cohesion of this new type of extract. The problems and the study protocol will be presented from the perspective of the international Document Understanding Conferences (DUC) and Text Analysis Conferences (TAC) campaigns for automatic multidocument summarization. A "topic" models users' information needs in the form of one or several topics that they want the summary to be about. The impact of using information from several documents (in contrast to one document) on the effectiveness of a summarization system will be studied.

4.1. Introduction

Multi-document summarization systems can be seen as the extension of several single-document summarizers in which the outputs are fused. However, as they work with several source documents, multidocument summarizers have a greater probability of producing redundant, incoherent and even contradictory information. It is important to understand that redundancy is not the only problem faced by multidocument summarizers: the temporal coherence of the events described in the documents is equally important.

Figure 4.1 shows a schematic description of an extractive multidocument text summarization system.

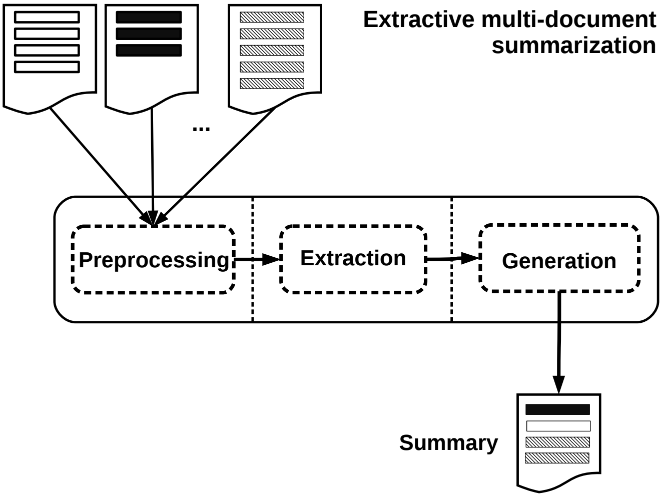


Figure 4.1. *Extraction based multidocument summarization system*

4.2. The problems of multidocument summarization

The generic task of multi-document summarization (MDS) consists of producing a unique summary from a collection of documents which are probably about the same topic. The first automatic MDS systems were developed by McKeown and Radev in the 1990s [MCK 95a, MCK 99, RAD 98]. The task of MDS has been greatly inspired by corresponding tasks at the DUC and TAC campaigns [1]. Therefore, huge advances have been made with regard to eliminating redundancy, chiefly with the works of Carbonell and Goldstein in 1998 [CAR 98], in which they introduced their method *Maximal Marginal Relevance (MMR)*.

[1] See DUC/TAC campaigns in section 8.6.

Most research into MDS [MAN 99a] apply statistical techniques to linguistic units such as words, sentences and paragraphs in order to choose, evaluate, classify and assemble them according to their relevance. This is done in a similar way to single-document summarization. Then, redundancy is identified so that any potential duplications and repetitions can be eliminated, while trying to keep the cohesion of the summary. However, in comparison to single-document summarization, new difficulties arise with MDS: clustering similar documents, designing and implementing antiredundancy measures, high compression rates meaning that sources are aggressively condensed, taking into account the temporality of texts and correctly resolving anaphoric references, among others [FUE 08].

In the context of MDS, [CAR 98, HAT 01, MCK 01, RAD 00] in particular have studied the clustering of similar documents. The problem of selecting the most salient fragments from each cluster, while guaranteeing the coherence of summaries, is the subject of intensive research. [SAG 04] developed a system in which the relevance of each sentence is measured in relation to its similarity to the centroid of a multidocument *corpus*. They combine the similarity to the centroid with the absolute position of the sentence in the document. They developed an antiredundancy method based on n-grams. Machine learning approaches have also been explored. For instance, [HIR 03]'s system has a similar approach to that used by [ISH 02] in single-document summarization. Thus, a support vector machine (SVM)[2] was trained on single-document data and was then used to rank the sentences in a multidocument set.

The problem of contradictory information is also present in MDS. In fact, the multiple documents could be written by different authors who have different styles and ways of structuring a document. Finding contradictory information about the same event is not rare and, worse still, facts and opinions can change over time. Therefore, documents

[2] "In machine learning, SVMs (also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis" (source: Wikipedia: http://en.wikipedia.org/wiki/Support_vector_machine).

written in different periods can contain conflicting information. Despite the efforts of the community, managing contradictory information remains a delicate and open problem.

4.3. The DUC/TAC tasks for multidocument summarization and INEX Tweet Contextualization

The problems of MDS were pinpointed at the National Institute of Standards and Technology (NIST)/DUC conferences from 2001 to 2007. The main task involved dealing with real and complex questions. The type of answer expected could not be a simple entity (a noun, a date or a quantity as classically defined in the *Text REtrieval Conference (TREC) Question-Answering* conferences[3]). DUC 2007 introduced a more complex task: evaluating MDS systems which identify updated information. This task will be discussed in section 4.6.

Formally, the problem of MDS can be expressed as follows: given a topic Q and a set of D relevant documents, the DUC task consists of generating a short, coherent and well-organized summary which answers the questions asked by the topic.

The topics are composed of two parts:

– the title of the topic;

– the narrative part containing the questions.

Each problem was created by an expert who brought together a group of connected documents and wrote a “topic” and a “description narrative”. For example, the 2006 and 2007 editions asked for 250 word summaries to be generated from groups of 25 documents. The documents were taken from the AQUAINT *corpus*: articles from the *Associated Press* and the *New York Times* from 1998 to 2000 and from the *Xinhua News Agency* from 1996 to 2000. An example of the topics and subtopics in documents D0603C and D0606F at DUC 2006 is presented in Table 4.1.

[3] http://trec.nist.gov/data/qamain.html.

Number and title (topic)	Question(s) (subtopics)
D0603C *Wetlands value and protection*	*Why are wetlands important?* *Where are they threatened?* *What steps are being taken to preserve them?* *What frustrations and setbacks have there been?*
D0606F *Impacts of global climate change*	*What are the most significant impacts said to result from global climate change?*

Table 4.1. *Examples of topics and subtopics in documents (D0603C and D0606F) at DUC '06*

[BAR 02] describe five representations of the topic, which they use in MDS. They also developed six algorithms for MDS that use different methods to extract information and weight sentences. The representations of the topic were based on information provided by a superficial semantic analysis of the document (*shallow parsing*).

Once the sentences are weighted and sorted, they must be selected and assembled to produce a unique multidocument summary. The length of the summary is set by the compression rate τ: the maximum number of words, percentage or number of sentences. At the DUCs, the summary produced could not contain more than 250 words. This is a challenge for systems, as they must pay particular attention to the choice of the final sentence: is it better to select the final sentence with a significant weight but exceed the limit of 250 words? Or it is better to include a sentence with a lower weight but which falls within the imposed limits? In fact, summaries which exceed the word limit are automatically truncated prior to evaluation, meaning that systems which choose the first alternative are penalized.

The systems which do best at the DUCs are almost entirely based on sentence extraction techniques, with a small amount of pre- and postprocessing [DAN 06, DAU 05]. Unfortunately, the granularity of sentence extraction systems is often insufficient for the production of short summaries from large sets of documents [DAU 06].

This is why approaches turned to sentence compression: a higher number of shorter sentences can be included in the summary within the same word limit [KNI 02, RIE 03, TUR 05].

The *Text Analysis Conference*s (TAC) have pursued MDS tasks by introducing extra difficulties. Thus, in 2008, the *Opinion Summarization Pilot Task* had the following objective: *[...] to generate well-organized and fluent summaries of opinions about specified targets, as found in a set of blog documents*[4]. In 2009, update summarization tasks and Automatically Evaluating Summaries of Peers (AESOP), applied to MDS, continued along this route. In 2010, the task of guided summarization made its entrance. It consisted of writing a 100 word summary from a set of 10 newswire articles about a given topic, where the topic was part of a previously defined category. There were five categories of topics:

– accidents and natural catastrophes;

– attacks;

– health and safety;

– endangered resources;

– investigations and trials.

In 2011, the new multilingual task was added: *Multi-Lingual Summarization*, whose main objective was to promote the use of multilingual algorithms for automatic MDS.

Also in 2011, the CLEF-INEX task, Question-Answering (QA) task and Tweet Contextualization Task introduced the double problem of short (less than 500 words) MDS, guided not by a query but by a tweet (of 140 characters or less). This tweet must be contextualized (see section 8.7).

After the generation of multidocument summaries, the problem of evaluating them arises. As explained in Chapter 2, manually or semi-automatically evaluating the quality of single-document summaries remains an arduous and costly task and automatic evaluation is still unreliable. In MDS, the problem is no easier.

The organizers of CLEF-INEX prioritized ranking participating systems on their informativity and readability (which were manually

[4] Source: http://www.nist.gov/tac/2008/summarization/op.summ.08.guidelines.html).

and automatically calculated, see section 8.7). During the DUC/TACs, manual and semi-automatic approaches were used to this end. PYRAMID [NEN 04, NEN 07] and BASIC ELEMENTS (BE) [HOV 06] were therefore employed. ROUGE measures [LIN 04] were retained for semi-automatic multidocument evaluation. Several other manual measures were also evaluated: coherence, grammaticality, non-redundancy, relevance in relation to the topic and linguistic quality. See section 8.6 for more details about these measures.

4.4. The taxonomy of multidocument summarization methods

Figure 4.2 shows the taxonomy, adapted from [LIT 10a], of statistical methods for sentence weighting independent of language. These methods were used in several automatic MDS systems [LIT 10b]. In all cases, s represents a sentence in the document D, composed of $i = 1, 2, \dots, \rho$ sentences; and $\omega(s)$ is its score (weight).

4.4.1. *Structure based*

These sentence weighting methods do not need to represent the text and are based exclusively on the structure of the document:

– Based on the position of the sentence [BAX 58]:

- $\omega(s) = i$: the score is high for sentences close to the end of the document;

- $\omega(s) = i^{-1}$: the score is high for sentences close to the beginning of the document;

- $\omega(s) = \max[i^{-1}, (\rho - i + 1)^{-1}]$: the score is high for sentences close to the borders of the document;

- $\omega(s) = (i - \rho \text{ MOD } 2)^2$: the score of a sentence is modeled by a parabola of its position [TOR 09]. The score is maximal at the borders;

– based on the sentence length [SAT 01]:

- $\omega(s) = |w_1, w_2, \dots, w_n|$: number of words in the sentence;

- $\omega(s) = |w_1| + |w_2| + \dots + [w_n|$: number of characters in the sentence.

4.4.2. *Vector space model based*

The vector space model (VSM) (see section 2.1.1 and Appendix A.2.1) is used to represent the text:

– Based on frequency:

- $\omega(s) = \max_{m \in \text{clusters}(s)} \{|\mathrm{K}(s)|^2 / |m|\}$: the Luhn algorithm [LUH 58];

- $\omega(s) = \sum_{i \in \mathrm{K}(s)} \mathrm{tf}_i$: the Edmundson algorithm [EDM 69]. Sum of the Term frequency (tf) frequency of keywords;

- $\omega(s) = |\mathrm{K}(s)| / |\mathrm{K}(D)|$: ratio of keywords in the sentence and in the document [FAT 04];

- $\omega(s) = (\sum_{j \in s} \mathrm{tf}_j) / |s|$: word frequency averaged over the ρ sentences [VAN 07];

- $\omega(s) = \sum_{j \in s} \mathrm{tf}_j \times \mathrm{isf}_j$; where $\mathrm{isf}_j = 1 - \log \frac{n_i}{N}$ and n_j is the number of sentences containing the word j. Word frequency multiplied by the inverse frequency of sentences, averaged over the ρ sentences [NET 00];

- singular value decomposition (SVD): the score of a sentence is calculated according to the length of the sentence vector $\Sigma^2 W_k^T$, after calculating the SVD of words of the sentence matrix $S = U_k \Sigma_k W_k^T$ [STE 04];

– based on the title T [EDM 69]:

- $\omega(s) = |s \cap T| / \min\{|s|, |T|\}$: overlap;
- $\omega(s) = |s \cap T| / |s \cup T|$: Jaccard similarity (see A.2.2);
- $\omega(s) = \cos(\vec{s}, \vec{T}) = \vec{s} \times \vec{T} / \|\vec{s}\| \|\vec{T}\|$: cosine similarity;

– based on Jaro-Winkler (JW) similarity of characters between the sentence s and a query Q [BOU 08a]:

- $\omega(s) = \mathrm{sim}_{JW}(\vec{s}, \vec{Q})$;

– based on document overlap [LIT 10a]. These methods weight a sentence according to its resemblance to the rest of the sentences in the complement document $D \backslash s$. They are based on the following idea: the more document content is found in a sentence, the more important it is. Sentences containing repeated information are not considered:

- $\omega(s) = |s \cap T| / \min\{|s|, |D \backslash s|\}$: overlap with the title T;
- $\omega(s) = |s \cap T| / |S \cup D \backslash s|$: Jaccard similarity with the title T;
- $\omega(s) = \cos(\vec{s}, \overrightarrow{D \backslash s}) = \vec{s} \times \overrightarrow{D \backslash s} / \|\vec{s}\| \|\overrightarrow{D \backslash s}\|$: cosine similarity.

4.4.3. *Graph based*

These methods use graphs (see section 3.5) for the representation of the text:

– the popularity of a vertex is used in place of the word frequency. Words are considered significant if they are represented by the vertices of a popularity which is higher than a predefined threshold [LIT 10a]:

- Luhn algorithm;
- graph extension of the ratio of keywords;
- graph extension of overlap;
- $\omega(s) = \sum_{i \in s} \text{popularité}_i / |s|$. Average popularity of all vertices;

– based on PAGERANK. The PAGERANK value of vertices is used in place of the word frequency. Words are considered significant if they are represented by vertices with a PAGERANK score which is higher than a predefined threshold:

- Luhn algorithm;
- PAGERANK of keywords [LIT 10a];
- PAGERANK of overlap [LIT 10a];
- LEXRANK of vertices (equation [3.9]);
- TEXTRANK of vertices (equation [3.10]);

– based on similarity [LIT 10a]:

- overlap of edges in the title T and the sentence s;
- Jaccard similarity between the edges in the title T and the sentence s;
- overlap of edges in the sentence s and the complement document $D \backslash s$;
- Jaccard similarity between the edges in the sentence s and the complement document $D \backslash s$;

– based on the textual energy E of the sentence s:

- $\omega(s) = |\sum_i E|$; $(E = S \times S^T)^2$ (equation [3.40]).

4.5. Some multi-document summarization systems and algorithms

In this section, some algorithms and systems used in multidocument text summarization will be presented: SUMMONS, MMR, a biographic summarizer, multidocument ENERTEX, MEAD, CATS, SUMUM, NEO-CORTEX, ICSI and CBSEAS.

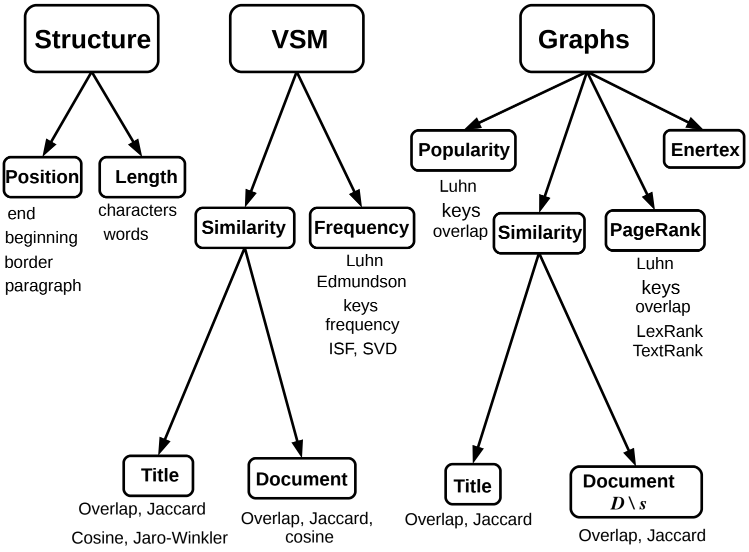


Figure 4.2. *Taxonomy of statistical methods for sentence weighting, independent of language*

4.5.1. SUMMONS

SUMMONS, built by Radev and McKeown [RAD 98], was one of the first multidocument systems. It is a summarization system based on knowledge. It produces summaries of a small number of news articles in the domain of terrorism. SUMMONS uses a set of semantic models extracted by an automatic understanding system. The techniques involve large amounts of knowledge engineering, even for the restricted domain of terrorism. [RAD 98]'s results indicate that SUMMONS generates informative and coherent multidocument summaries which cover different aspects of the domain. However, this method has several disadvantages. It requires a manually generated knowledge model, an information extraction system which is specific to the topic and their representation and empirical methods to find the possible relations between the extracted models. Therefore, SUMMONS is not suitable for analyzing documents outside of the domain.

4.5.2. *Maximal marginal relevance*

A summarizer finds itself faced with the task of generating an extractive text summary exclusively by selecting sentences. Imagine that the summarizer has processed (read) the topic, the narrative description and has browsed through all the documents. The summarizer will perhaps choose a sentence which responds well to the query and consider it for the summary. What should it do to select a second sentence? A good summarizer wants to respond to the query, but it does not want the second sentence to contain the same information as the first sentence it has already chosen. After all, the summarizer is working within the constraint of summary length.

The MMR method [CAR 98, GOL 00] is an algorithm which makes the following idea a reality: maximizing relevant information while minimizing redundancy. Initially developed for information retrieval, the MMR algorithm has been successfully applied to the task of MDS. The equation for weighting a sentence s according to MMR is:

$$\omega_{\text{MMR}}(s) = \arg\max_{s \in D \setminus \text{Sum}} \left[\, \lambda \underbrace{\text{sim}_1(s, Q)}_{\text{Relevance}} - (1-\lambda) \underbrace{\arg \max_{s_\mu \in \text{Sum}} \text{sim}_2(s, s_\mu)}_{\text{Redundancy}} \,\right] \qquad [4.1]$$

where D is the set of sentences in the document, Sum is the set of sentences already selected for the summary, Q is a query (topic of the summary), λ is an interpolation coefficient and $\text{sim}_1(\bullet)$ and $\text{sim}_2(\bullet)$ are two functions which return the similarity between two sentences.

At each stage, the MMR score contains the sentence with the highest score out of the remaining sentences: by maximizing its similarity in relation to the query and by penalizing its resemblance to its closest neighbor in the set of selected sentences. With $\lambda = 1$, the algorithm chooses all the sentences concerned, without worrying about redundancy. With $\lambda = 0$, relevance is not given importance and the sentences selected are as dissimilar as possible (by maximizing coverage). To create a summary with the MMR algorithm, sentences are selected and concatenated until the limit of summary length is reached. The ideal values λ seem to vary considerably according to the source of documents. To characterize the MMR algorithm, it is necessary to define its similarity function. The normalized similarity function, also borrowed from information retrieval, is the cosine distance between the word vectors, for each sentence, where the value of each word is tf.idf. In other experiments, [GIL 11] obtained an equivalent performance when using a simple word overlap measure: the number of common non-stop words, normalized by the length of the longest sentence.

4.5.3. *A multi-document biography summarization system*

Task 5 at the DUC 2004 (see section 8.6) consisted of generating short summaries (665 bytes) guided by a question of the type *Who is X?*. An example of these questions is: “Who is Charles Bukowski?”. An important application of this task consists of creating biographies. However, manually generating biographies is tiresome when information relating to the person in question is hidden among other not obviously related information. In response to this problem, [DAU 02, HAR 03, MCK 01, ZHO 05b] developed several algorithms for biographic summarization. [ZHO 05b]’s multidocument approach will be presented. This system uses two classification modules:

– a fine classification containing 10 categories: given one or several texts about a person, this module classifies the sentence s into one of the 10 predefined categories [5];

– a binary classification: this module makes a decision in order to find out whether the sentence s should belong (*biography* class) in the summary or not (*none* class).

[5] Nine categories contain biographic elements and the tenth contains no biographic information: *biography, fame, personality, social, education, nationality, scandal, personal, work, none.*

To construct these modules, three learning algorithms were used: a naïve Bayes method (see section 3.2.1), A Library for Support Vector Machines (LIB-SVM) software[6] and a C4.5 decision tree algorithm. The informativeness of each word is calculated using the itf (inverse term frequency, see Appendix A.2.1, (equation [A.5])) of the DUC documents and the TREC-9 *corpus* (approximately 400 million words):

$$\text{info}(w) = \text{itf}_{\text{DUC}}(w)/\text{itf}_{\text{TREC}}(w) \qquad [4.2]$$

where $\text{itf}_{\text{DUC}}(w)$ is the itf subset of the DUC documents of the word w, and $\text{itf}_{\text{TREC}}(w)$ the itf subset of the TREC-9 *corpus* of the word w. Weighting a sentence of N words is obtained by:

$$\omega(s) = \sum_{i=1}^{N} \text{info}(w_i)/|s| \qquad [4.3]$$

Redundancy is reduced as follows: from the complete summary, sentences are removed one by one so that the semantic similarity of the remaining summary in relation to the original text remains stable [MAR 99]. The architecture of this algorithm is shown in Figure 4.3. The results of task 5 at DUC 2004 ranked this algorithm among the top algorithms in terms of ROUGE-L (the longest subsequence [LIN 04]). Figure 4.4 presents an example of a biographic summary generated by this system.

4.5.4. *Multi-document* ENERTEX

The ENERTEX similarity measure has been used to weight sentences and obtain generic single-document summaries (see section 3.9.3). An elementary modification, introducing a topic, enables the generation of multidocument summaries guided by users' information needs [FER 08a].

[6] Software available at: http://www.csie.ntu.edu.tw/cjlin/libsvm.

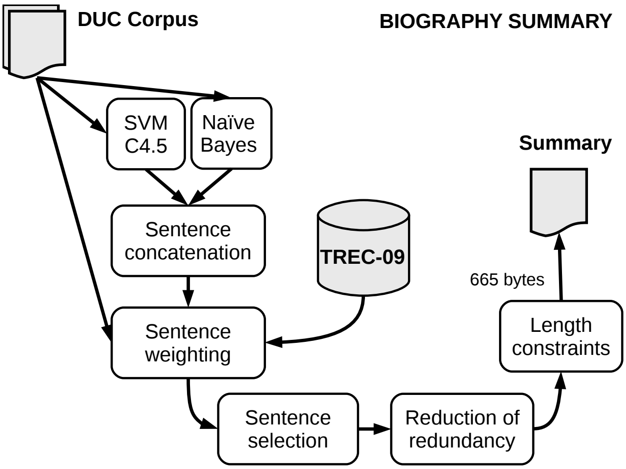


Figure 4.3. *Multi-document biography summarization system created by [ZHO 05b]*

Question: *Who is Sir John Gielgud?*

Response: *Sir John Gielgud, one of the great actors of the English stage who enthralled audiences for more than 70 years with his eloquent voice and consummate artistry, died Sunday at his home Gielgud's last major film role was as a surreal Prospero in Peter Greenaway's controversial Shakespearean rhapsody.*

Figure 4.4. *Example of output from a biography summarizer created by [ZHO 05b]*

Corpora from the main DUC tasks from 2005 to 2007 were used to test ENERTEX in multidocument mode. Given 45 topics with 25 groups of documents, ENERTEX was required to produce fluent 250 word summaries responding to the questions asked by the topics. The set of 25 documents was concatenated into one document D of ρ sentences in chronological order. The topic was added to D as the final sentence ρ+1. A standard preprocessing operation was applied to obtain the sentence-term matrix (see section 2.1.1).

The textual energy between the sentence-topic and the other sentences in the document D was calculated. The summary is generated with sentences with absolute maximum energy values.

ENERTEX's antiredundancy strategy is not linguistic. It compares energy values between sentences before including them. Sentence length is used to eliminate sentences which are too long (those which exceed the summary limit of 250 words) or too short [FER 08a] avoided redundancy in the following way: the most energetic sentence is added to the summary in turn. A candidate sentence i will be part of the summary, if for all the k sentences already present in the summary:

$$|E_i - E_j| \geq \epsilon\,;\ j = i+1, \dots, k \qquad [4.4]$$

A threshold $\epsilon = 0.003$ which corresponds to sentences with one or two different words was identified empirically.

The results from the DUC data from 2005 to 2007 [FER 08a] show that ENERTEX was ranked among the top participating systems. Figure 4.5 shows where the ENERTEX (triangle) system was ranked in the ROUGE evaluations at DUC 2007 in comparison to other participants and baselines. In 2007, two baseline evaluations were included. The first evaluation is the same as in 2005 and 2006: a random baseline by random extraction of sentences. The second evaluation is a more *intelligent* baseline, produced by a generic summarization system. This explains why this baseline outperformed almost half of the participating systems.

4.5.5. MEAD

The topic-oriented MDS system (MEAD) is a free software platform for generating automatic text multidocumentsummaries[7].

[7] MEAD was created by several people: Dragomir Radev, Simone Teufel, Horacio Saggion, Wai Lam, Elliott Drabek, Hong Qi, Danyu Liu, John Blitzer and Arda Çelebi, during eight weeks of the summer of 2001 at Johns Hopkins University.

Originally programmed to process two languages – English and Chinese – over time MEAD has become multilingual. MEAD implements different algorithms for sentence extraction: based on the position, the {1,2}-gram-overlap, the centroid of documents and on the cosine (tf.idf) applied to a query. MEAD also implements evaluation methods (MEADEval): *MEADEval currently supports two general classes of evaluation metrics: co-selection and content-based metrics. Co-selection metrics include precision, recall, Kappa, and Relative Utility, a more flexible cousin of Kappa* [RAD 00, RAD 01a, RAD 03].

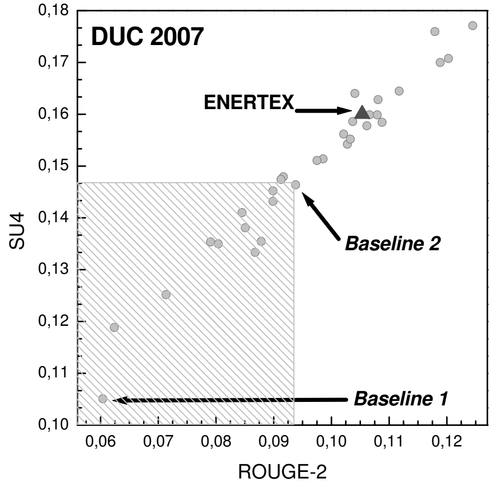


Figure 4.5. ROUGE-2 *evaluation versus* SU4 *of systems participating at DUC '07*

Figure 4.6 presents the basic idea of MEAD: if cosine similarity α between *document*$_1$ and a centroid is less than a threshold, then *document*$_1$ will be included in the representative group of the centroid. The most important k words taken from all documents are the words which compose the centroid.

The documents are represented using their tf.idf weighted vector. The CIDR algorithm generates a centroid using only the first document in the group. In Radev *et al.*'s words: *A centroid is a set of words that are statistically important to a cluster of documents* [RAD 04]. The tf.idf values of new documents are compared with the centroid using the similarity equation [4.5].

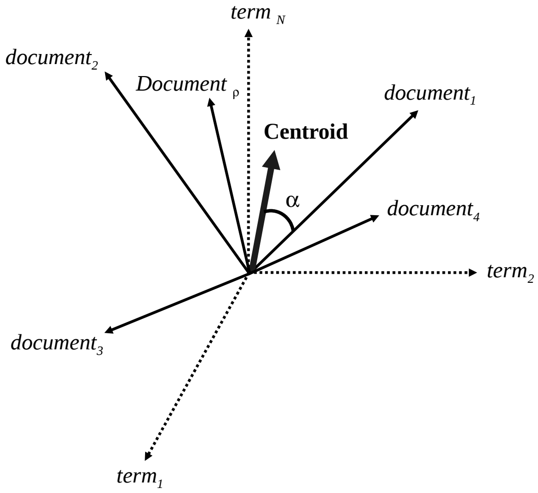


Figure 4.6. *Conceptual representation of* MEAD*, in a space of N words, where there are* ρ *documents and their centroid*

If the similarity measure $\text{sim}(\vec{D}, \vec{C})$ is less than a threshold, the new document D is included in the group C:

$$\text{sim}(\vec{D}, \vec{C}) = \frac{\sum_k (D_k \times c_k \times \text{idf}(k))}{\sqrt{\sum_k (D_k)^2}\sqrt{\sum_k (c_k)^2}} \qquad [4.5]$$

where D_k is the document vector and c_k the centroid vector.

MEAD's weighting function weights a sentence s based on a linear combination of characteristics (the position of s, the centroid and the similarity):

$$\omega(s) = \lambda_p P(s) + \lambda_c C(s) + \lambda_f F(s) \qquad [4.6]$$

where $\lambda_{p,c,f}$ are the weights of the position, the centroid and the similarity, respectively.

LEXRANK (see section 3.5.4.1) was incorporated into the MEAD system and obtained very good results on the DUC *corpora* from 2003 and 2004 [ERK 04].

Several systems were built with MEAD as the summarization core, including NEWSINESSENCE [RAD 01b, RAD 01d], system which retrieves online news and generates their summaries and WEBINESSENCE [RAD 01c] which conducts a clustering operation and summarizes web search results.

MEAD sources are publicly available at: http://www.summarization.com/mead/. A demo of the Dutch version is available at: http://www.cnts.ua.ac.be/iris/sumdemo.html.

4.5.6. CATS

Cats is an Answering Text Summarizer (CATS) is an MDS system guided by a topic, developed at the University of Montreal [FAR 05]. It produces a summary from a set of documents about the same topic, which incorporates users' information needs expressed in the form of questions. It starts by analyzing the topic and the documents to identify a list of text segments containing interesting aspects in relation to the topic. It then compares these segments with those identified in the questions. CATS obtained very good results at the DUC 2005 campaign. Figure 4.7 shows the general architecture of the CATS system.

The DUC 2005 task consisted of generating a summary of less than 250 words from a set of 25–50 newspaper articles about the same topic. The resulting summary had to deal with a particular aspect of this topic, which was identified by a question written over several lines. The topic indicates the desired granularity of the summary:

– specific for describing precise entities and/or events;

– general for generalizing facts.

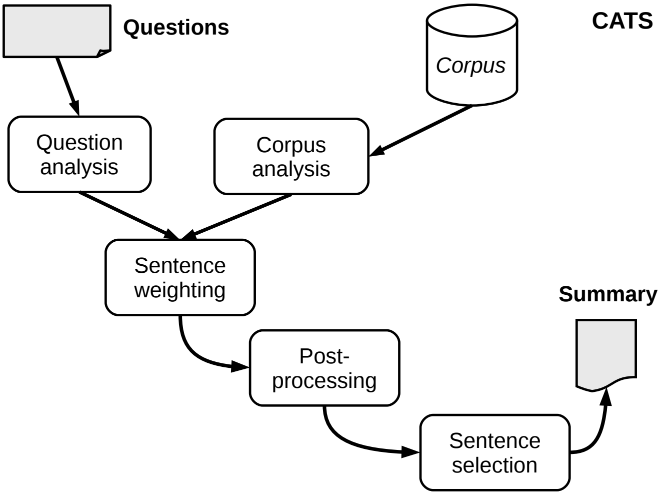


Figure 4.7. CATS*, multidocument summarization system guided by a topic*

CATS analyzes questions in two stages: it identifies the type of named entities and splits sentences into elementary elements. Scoring and selecting sentences are at the core of CATS. Cosine similarity with an empirical threshold enables interesting topic segments to be identified. The calculation is based on the title of the question (which for the most part represents the general subject) and the topic. A score is attributed to each sentence in the *corpus* associated to the question. This score is a linear combination of seven normalized measures between 0 and 1 [FAR 05]:

– BASIC ELEMENTS: comparing the BASIC ELEMENTS of the sentences in the question with the BASIC ELEMENTS of the sentences in the corpus;

– cosine similarity: calculating the cosine similarity between the sentences in the question and the sentences in the *corpus*;

– sentence weighting: the sum of the weight of words obtained from the idf;

– absolute position: score based on the position of the sentence in the text;

– relative position: score based on the position of the sentence in the paragraph;

– named entity: counting the number of named entities in the sentence of the *corpus* which are in the same category as a named entity in the question;

– prototypical expressions: the number of these expressions (for instance: *as a consequence, as a corollary, as a logical conclusion, as a matter of fact, as a result, as against, as evidence, as far as*) is counted. This indicates the sentences with the greatest probability of containing important information (for instance the sentences in the conclusion).

To obtain a summary which is both concise and coherent, some postprocessing operations are carried out on sentences to eliminate the least important parts or to replace certain expressions with other, shorter expressions. The two postprocessing operations which are applied are the resolution of temporal expressions and sentence compression. The resolution of temporal expression is realized through a specialized module (TempEx), which eliminates certain expressions such as *several days* or *weekly* and normalizes the days of the week to a standard format, MM/DD/YY, using specific markers. Sentence compression eliminates all the text between **(), [], {}, –** and **–**. For summaries requiring a general level of granularity, the Collins parser [COL 03] as well as TREETAGGER are used. The syntactic tree is analyzed and the branches corresponding to subordinate clauses starting with *who, when, where and which* are eliminated. This operation enables descriptions of people and places to be deleted, which are often have no use in summaries with a general level of granularity [FAR 05].

4.5.7. SUMUM *and* SUMMA

The *Summarization at Université de Montréal* (SUMUM) system [SAG 02b] is a multidocument summarizer which produces indicative and/or informative summaries of scientific and technical documents. SUMUM is based on a linguistic and conceptual model of technical articles:

author ▶ article ▶ problem ▶ results ▶ introduce the topic ▶ define the topic ▶ develop the topic ▶ etc.

SUMUM realizes the selection and presentation of information in a similar manner to professional summarizers. Informative summaries are generated as follows: first, the reader sees an indicative summary which identifies the topics in the document (introducing the authors, the discussion section, etc.), then, if the reader is interested in certain themes, specific information about the topics are presented as an informative summary (from the source document). SUMUM generates summaries via the following stages:

– text analysis via automata and a dictionary of concepts;

– indicative selection, using regular expressions, frames guided by the title and words;

– generation of the indicative summary with several sentence transformations;

– informative selection with dynamic correspondence (using words);

– generation of the informative summary, including a presentation of additional sentences according to the user's interest in precise terms.

The SUMUM algorithm is similar to summaries based on *concept-based abstracting* (CBA) [PAI 93]. CBA has been used to produce the summaries of technical articles in specific domains such as agriculture. The semantic roles and the high- and low-level properties are first identified by manually analyzing a *corpus*, then the patterns are specified while the stylistic regularities of the expression of semantic roles in the texts are taken into account. SUMUM is a system which is not limited to the simple extraction of passages, but which seeks to generate an informative and coherent abstract (see Chapter 7).

Figure 4.8 shows the simplified architecture of the SUMUM system. For more details, see [SAG 02b].

Figure 4.9 shows an example [SAG 00b] of indicative and informative summaries produced by the SUMUM system.

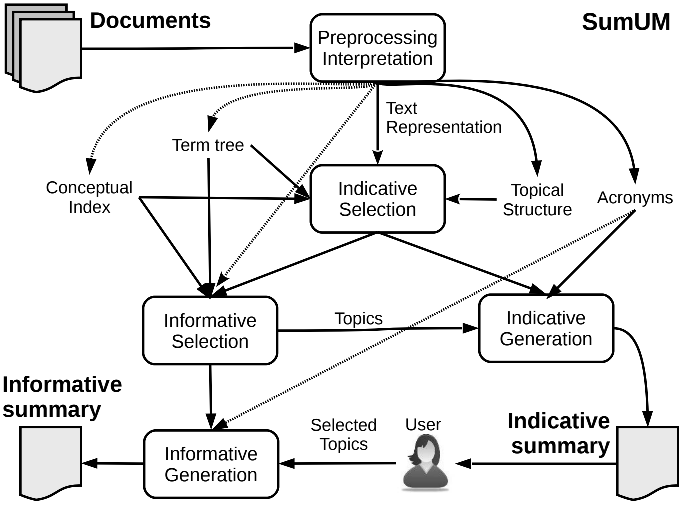


Figure 4.8. *Simplified architecture of the multidocument summarization system* SUMUM

SUMMA [SAG 08, SAG 14] is a set of text summarization resources freely available for the development of text summarization applications [8]. It has been developed in Java using the GATE Java library [MAY 02] and relying on GATE documents, GATE annotations, and other data-structures for information storage and retrieval. Over twenty implemented algorithms are distributed in SUMMA, allowing the user to create different summarization applications by combining components in a pipeline. These algorithms implement well known single-document, multi-document, and guided summarization methods based on sentence scoring and extraction such as the position method, the term frequency method, the cuephrase method, the title method, the centroid-based method, etc. Most SUMMA components are based on statistical computations based on the Vector Space Model which has been discussed in this book [9].

[8] http://www.taln.upf.edu/pages/summa.upf

[9] We thank Horacio Saggion for providing this description.

4.5.8. NEO-CORTEX

NEO-CORTEX [BOU 07a] is the multidocument version of CORTEX [TOR 02] [10]. NEO-CORTEX maximizes the number of words in the summary while prioritizing the most informative segments. Intrasummary redundancy is minimized by previously applying a threshold of intersentence similarity. Countless constraints are faced while segments are concatenated. Indeed, segments must be temporally ordered in the summary, i.e. the publication dates of the source documents must be respected as well as the position of a sentence in a document, if two sentences come from the same source. Two parameters were introduced to obtain coherent summaries from several document sources: a global parameter – similarity between a document and the topic – and a local parameter – word overlap between a sentence and the topic. Finally, a surface postprocessing operation is applied to the sentences in the summary.

Global similarity

Sentence scores are calculated in each document. They are normalized to highlight the degree of relevance of documents in relation to the topic. Indeed, a relevant sentence in a document can have a lower score than an irrelevant sentence in another document. This is due to the interdocument independence of the scores calculated.

A document D is represented by $\vec{D} = (\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_\rho)$; ρ is the total number of sentences and a topic represented by $\vec{Q} = (q_1, q_2, \ldots, q_N)$ words; N is the size of the lexicon (documents plus the topic). Cosine similarity $\cos(\vec{D}, \vec{Q})$ is calculated between the document vectors $\vec{D}$ and the topic $\vec{Q}$. The tf.idf weight of words was calculated from the set of documents from the DUC 2007 collection. Similarity is normalized between $[0, 1]$.

[10] The algorithm for single-document summarization is presented in section 3.7.

Article: ***Features 3D scanning systems for rapid prototyping, Jim Clark, Assembly Automation, Vol 17 Issue 3, 1997*** (97 phrases)

Indicative summary:
Describes two non-contact scanning systems, REVERSA and ModelMaker. Topics: CADAM system; ModelMaker; REVERSA; standard dual view system; system

Informative summary (by expanding REVERSA and ModelMaker):
The first sensor, REVERSA, is a standard dual view system which can be mounted on a purpose built four-axis ¯translation stage or retro-ffitted to third party CNC and CMM (the coordinate measuring machine) machines. REVERSA is a dual viewpoint non-contact laser scanner which comes complete with scanning software and data manipulation tools. The ModelMaker scanning system is a combination of a 3D laser stripe sensor, 6 DoF position localizer and a PC in which up to two interface cards are mounted. ModelMaker can simply be retro fitted to existing arms providing the benefits of a portable CMM with dense depth datasets. Use of ModelMaker allowed the part to be scanned in under one hour, a job that might have taken days using the traditional point to point touch process, and did deform the model in the process.

Figure 4.9. *Example of indicative and informative summaries produced by* SUMUM *[SAG 00b]*

Local overlap

Sentences extracted to constitute the summary must have a certain amount of information in common with the topic. To estimate this amount, the number of common words between the topic and each sentence s was calculated. The overlap r of each sentence is the normalized cardinality between $[0, 1]$ of the intersection between the set of words in each sentence s and the set of words in the topic Q:

$$r(s, Q) = \frac{|s \cap Q|}{|Q|} \qquad [4.7]$$

$|\bullet|$ is the cardinality of the set $\bullet$. This measure improves the weighting of sentences which contain words from the topic and reduces the problem of heavily weighted sentences which contain no word from the topic.

Sentence weighting

The score $\omega(s)$ of the sentence s of a document $\vec{D}$ given a topic $\vec{Q}$ is the linear combination:

$$\omega(s) = \lambda_0 \textsc{Cortex}(s) + \lambda_1 \cos(\vec{D}, \vec{Q}) + \lambda_2 r(s, Q)\,;\ \sum_i \lambda_i = 1 \quad [4.8]$$

λ_i are empirical parameters whose sum is worth 1. $\textsc{Cortex}(\bullet)$ is the weight obtained with the decision algorithm (equation [3.33]), $\cos(\bullet)$ is cosine similarity and $r(\bullet)$ overlap (equation [4.7]). The summary is generated with a compression rate τ set by the user (a number or percentage of sentences).

Postprocessing

A simple surface-level postprocessing operation is carried out, which considers the order the sentences appear in the summary:

– rewriting acronyms: definitions are automatically identified via regular expressions;

– normalizing temporal references: dates written as “December 21, 1980” are replaced with “21/12/1980”;

– rewriting ambiguous temporal references: “... by the end of next year, ...” using the tag **2010_10_10** becomes “... by the end of 2011 ...”;

– partial elimination of discursive structures: “But, she is ...” becomes “She is ...”;

– elimination of text contained between parentheses;

– normalizing punctuation (spaces and repeated punctuation, etc.).

Figure 4.10 shows the general architecture of the MDS system NEO-CORTEX.

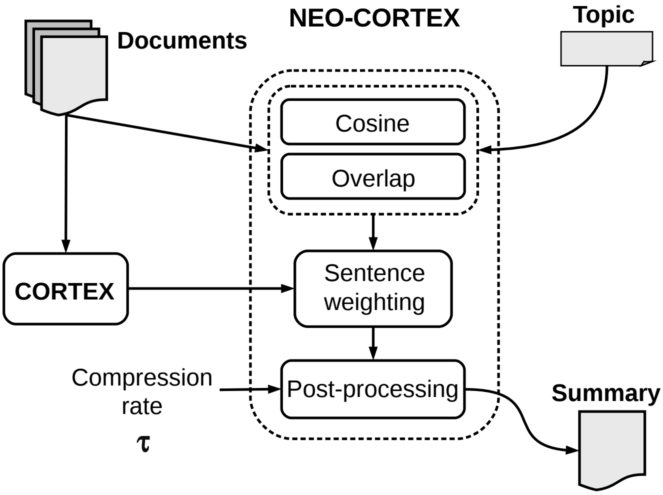


Figure 4.10. *General architecture of the summarization system* NEO-CORTEX

4.6. Update summarization

Update summarization is a task which is supposed to generate a multidocument summary by identifying new information relating to a topic. The objective is to give the user, who has previously read some of the documents (and probably their summaries as well), new facts while avoiding repeating information already included in previous summaries. This task was introduced as a pilot task at DUC 2007 and then became the main task at TAC 2008 and 2009.

4.6.1. *Update summarization pilot task at DUC 2007*

This task required short (approximately 100 words) multidocument summaries and their updates, according to a particular topic, to be produced, based on the assumption that the user had previously read some of the documents [DAN 07]. Therefore, the aim of the new summaries is to inform the reader of new information about a given topic. The document *corpus* for the pilot task came from AQUAINT. This *corpus* consisted of newswire articles from the *Associated Press*, the *New York Times* (1998–2000) and the *Xinhua News Agency* (1996–2000). There were 10 evaluators in total. Each topic was developed by four NIST evaluators.

The task was set out as follows. Given a topic and three groups of documents A, B and C, summarizers were required to create the following types of fluent summaries:

– a summary R_A from documents in set A;

– an update summary R_B of documents in B, supposing the reader had already read the documents in set A;

– an update summary R_C of documents in C, supposing the reader had already read the documents in sets A and B.

The clusters of documents had to be processed in chronological order. i.e. the documents in group B or C could not be accessed to produce the summary of group A; similarly, the documents in C could not be accessed to produce the summary of B. However, documents within a group could be processed in any order. Summary R_A summarized ten documents, summary R_B eight documents and summary R_C seven documents. The dates of the documents were as follows: date(*A*) < date(*B*) < date(*C*). Summaries were limited to 100 words (tokens demarcated by spaces). Automatic summaries exceeding this limit were truncated and no bonus was awarded to summaries shorter than 100 words. Twenty-two teams and two baselines took part in this task: a simple baseline and a generic CLASSY baseline (one for each group A, B or C).

4.6.2. *Update summarization task at TAC 2008 and 2009*

The update summarization task at TAC 2008 [DAN 08] was composed of two parts: to generate a first main summary and then an update summary. The document *corpora* came from AQUAINT-2 (a subset of *LDC english Gigaword* amounting to approximately 2.5 Gb of text – about 907,000 documents – from the period October 2004 to

March 2006). The articles were in English and came from a variety of sources, notably the *Agence France-Presse*, the *Central New Agency* (Taiwan), the *Xinhua News Agency*, the *Los Angeles Times-Washington Post*, the *New York Times* and the *Associated Press*. Forty-eight topics were developed by eight NIST evaluators, who also selected 20 relevant documents for each topic. The documents were collected in chronological order and divided into two sets, *A* and *B*, with 10 documents in each. The documents in group *A* preceded the documents in group *B* in temporal order.

The summarization task was the same for each summarizer (either a person or a system): given a topic and a set of documents about the topic, the task consisted of creating two short well-organized summaries of *A* and *B*, responding to the information needs expressed by the topic. Summary *A* concerned the first 10 documents and summary *B* the second 10, following the hypothesis that the reader had already read the documents in set *A*. The summaries were limited to 100 words: summaries exceeding this limit were truncated and no bonus was awarded for creating shorter summaries. Thirty-three teams took part in the task and 71 executions were sent in total (up to three submissions for each participating team). Manual evaluations were only expected for the first and second executions (57 in total) while automatic evaluations were realized for the set of submissions.

Table 4.2 shows examples of documents (D0842G and D0820D), with their titles and corresponding narratives for TAC 2008

The baseline summary in 2008 was defined as follows: extract the first sentence(s) in the most recent document and the summary produced is limited (truncated) to 100 words maximum. In TAC 2009, three baselines were implemented:

1) baseline$_1$ extracts all the leading sentences (up to 100 words) in the most recent document;

2) baseline$_2$ is a copy of one of the reference summaries but with the sentences ordered at random;

3) baseline$_3$ is a summary containing sentences which have been manually extracted from the *corpus*.

Number	D0842G
Title	Natural Gas Pipeline
Narrative	*Follow the progress of pipelines being built to move natural gas from Asia to Europe.* *Include any problems encountered and implications resulting from the pipeline construction.*
Number	D0820D
Title	Submarine Rescue
Narrative	*Describe efforts of the Russian navy to rescue the trapped submariners and any assistance provided by other countries. Include information regarding the results of the rescue mission and the results and consequences of the subsequent investigation into the matter.*

Table 4.2. *Some examples of topics, titles and narratives (TAC '08)*

Basically, the 2009 task repeats the TAC 2008 update summarization task. Manual evaluations in 2008 were realized by measuring the overall responsiveness on a scale of 1 (*very poor*) to 5 (*very good*); overall readability was measured on the same scale of 1 to 5 and with the Pyramid metric [NEN 04]. In 2009, overall responsiveness was evaluated on a 10-point scale rather than a 5-point scale [11]. Automatic evaluations were realized using Rouge-2 and -SU4 [LIN 04] and Basic Elements (BE) [HOV 06]. In both cases the *Jackknifing* method was used [12] for these metrics:

– to evaluate each reference model in relation to three other remaining models;

– to evaluate each automatic summary four times, each time in relation to a set of three summaries with different references.

[11] http://www.nist.gov/tac/2009/Summarization/update.summ.09.guidelines.html.

[12] "In statistics, the jackknife is a resampling technique especially useful for variance and bias estimation. The jackknife predates other common resampling methods such as the bootstrap. The jackknife estimator of a parameter is found by systematically leaving out each observation from a dataset and calculating the estimate and then finding the average of these calculations. Given a sample of size N, the jackknife estimate is found by aggregating the estimates of each $N - 1$ estimate in the sample" (source: Wikipedia: http://en.wikipedia.org/wiki/Jackknife_(statistics)).

Table 4.5 shows a reference summary and two automatic summaries, taken from the TAC evaluation of the update summarization task.

4.6.3. *A minimization-maximization approach*

An automatic multidocument summarizer guided by a topic to identify new information will be presented. The system is based on a minimization-maximization approach which is based on two concepts: the elimination of redundancy by limiting information that is repeated in the generated summary and the previous summaries; novelty detection [BOU 07b, BOU 09a].

4.6.3.1. *Relevance to the topic*

A standard preprocessing operation is applied to the *corpus*: documents are split into sentences; sentences are normalized, filtered (function words and too-common words are eliminated) and meaningful words are stemmed with the Porter algorithm [POR 80]. The relevance between a sentence $\vec{s}$ and the topic $\vec{Q}$ is calculated using two similarity measures: cosine and Jaro-Winkler distance [13]. When certain errors occur during preprocessing, the Jaro-Winkler distance enables scores to be smoothed:

$$\text{pertinence}(\vec{s}, \vec{Q}) = \lambda \cos(\vec{s}, \vec{Q}) + (1 - \lambda)\text{sim}_{JW}(\vec{s}, \vec{Q}) \qquad [4.9]$$

where $\vec{Q}$ is the topic vector (query), $\vec{s}$ is the sentence and $\text{sim}_{JW}(\bullet)$ is the Jaro-Winkler similarity between the sentence and the topic. λ is an empirical parameter evaluated at 0.7 on the DUC 2005 and 2006

[13] "[...] the Jaro-Winkler distance [WIN 90] is a measure of similarity between two strings. It is a variant of the Jaro distance metric [JAR 89], a type of string edit distance, and mainly used in the area of record linkage (duplicate detection). The higher the Jaro-Winkler distance for two strings is, the more similar the strings are." (source: Wikipedia: http://en.wikipedia.org/wiki/Jaro-Winkler_distance).

corpora. The smaller the angle α, the higher the relevance; see Figure 4.11.

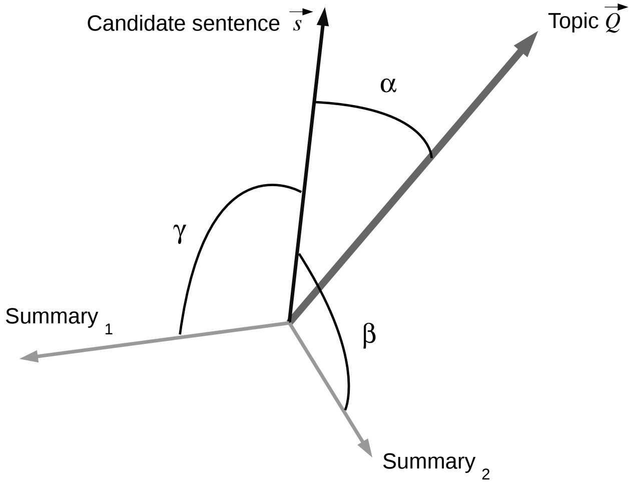


Figure 4.11. *Minimization-maximization between $\vec{s}$, $\vec{Q}$ and the previous summaries*

4.6.3.2. *Eliminating redundancy*

A system which identifies new information must deal with two different redundancy problems: redundancy within each summary and redundancy between summaries. The first concerns identifying duplications. To deal with this, the system measures similarity between sentences which are already part of the summary and candidate sentences. The latter are eliminated if their similarity is higher than an empirically fixed threshold.

The second type of redundancy is specific to the task: a summary is generated with the assumption that other summaries have previously been produced. As a result, the summary must contain additional information to inform the reader about new facts. The n_r first summaries are represented as a set of vectors $\Pi = \{\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_{n_r}\}$.

Redundancy is calculated by the angle between the sentence $\vec{s}$ and the n_r previous summaries:

$$\text{redundancy}(\vec{s},\Pi) = \sqrt{\sum_{i=1}^{n_r} \cos(\vec{s},\vec{r_i})^2}; \ 0 \leq \text{redundancy}(\bullet) \leq 1 \quad [4.10]$$

[NEW 04]'s work confirms that cosine similarity performs well with regard to the elimination of redundancy.

4.6.3.3. *Sentence weighting*

The minimization-maximization method calculates the ratio between relevance and redundancy:

$$\omega(\vec{s},\vec{Q},\Pi) = \frac{\text{relevance}(\vec{s},\vec{Q}\ )}{\text{redundancy}(\vec{s},\Pi)+1} \quad [4.11]$$

the lowest relevance value$(\vec{s},\vec{Q}\ )$ and the highest redundancy value$(\vec{s},\Pi)$ maximize the weight:

$$\omega(\bullet)\colon \begin{cases} \max\ \text{relevance}(\bullet) \\ \min\ \text{redundancy}(\bullet) \end{cases}$$

The sentence with the highest weight $\omega(\bullet)$ is the most relevant to the topic (relevance $\rightarrow$ 1) and, simultaneously, the most different in view of the Π previous summaries (redundancy $\rightarrow$ 0). The summary is constructed by concatenating the sentences with the highest scores according to the compression rate τ. Consequently, the final sentence has a high probability of being truncated.

A heuristic was developed to select this sentence and improve the readability of the summary. This technique is only applied if there are more than five remaining words otherwise a summary of non-optimal length will be generated. Grammaticality is protected by only eliminating function words and words with a very high frequency. Fifty or so regular expressions and an antidictionary were created for this selection [BOU 07a].

Table 4.3 shows the evaluation results for the automatic summarization system by minimization-maximization for the pilot task at DUC 2007.

Evaluation	Score	Rank (out of 24)
Quality of content	2.630	7
ROUGE-2	0.094	4
ROUGE-SU4	0.131	5
BASIC ELEMENTS (BE)	0.055	4
PYRAMID	0.273	5

Table 4.3. *Results of the minimization-maximization algorithm evaluated during the pilot task at DUC '07*

Figure 4.12 shows an overview of the architecture of this minimization-maximization system.

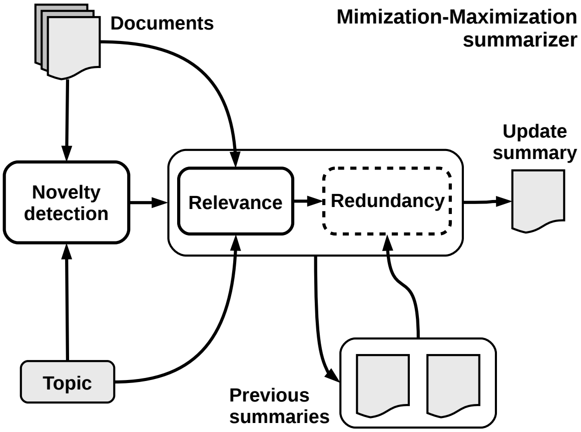


Figure 4.12. *General architecture of the minimization-maximization summarization system which identifies new information*

A variant of this system was named *A Scalable MMR Approach for Multi-Document Update Summarization* (SMMR), a reinterpretation of

MMR [BOU 08a, BOU 09a]. Thus, equation [4.1] was modified as follows:

$$\textsc{Smmr}(s) = \text{sim}_1(\vec{s}, \vec{Q}) \cdot [1 - \underset{s_h \in \Pi}{\arg\max}\ \text{sim}_2(s, s_h)]^{|\Pi|^{-1}} \quad [4.12]$$

where Π is the set of previously read (historic) summaries, $\text{sim}_1(\bullet)$ is relevance (equation [4.9]) and $\text{sim}_2(\bullet)$ is a normalized function of the longest common subsequence (LCS) between sentences s and s_h.

Table 4.4 shows the performance of SMMR during the main task (sets *A* and *B*) at TAC 2008.

Evaluation	Set A		Set B	
	Score	Rank	Score	Rank
Responsivity	2.417	26 out of 58	2.250	16 out of 58
Linguistic quality	2.458	22 out of 58	2.833	9 out of 58
PYRAMID	0.215	30 out of 58	0.215	30 out of 58
ROUGE-2	0.081	36 out of 72	0.068	43 out of 72
ROUGE-SU4	0.120	31 out of 72	0.112	32 out of 72
BASIC ELEMENTS (BE)	0.046	35 out of 72	0.046	35 out of 72

Table 4.4. *Results of the* SMMR *algorithm (set A on the left and set B on the right) for the main task at TAC '08 [BOU 08a]*

4.6.4. *The ICSI system at TAC 2008 and 2009*

The most highly performing system in terms of ROUGE evaluations during the two tasks at the TAC 2008 and 2009 competition was [GIL 08]'s from the *International Computer Science Institute (ICSI)*, at Berkeley. At TAC 2009, it also received the best overall scores for the main task of MDS and was ranked second in the update summarization task. In 2008, [GIL 08] developed the first version of an approach based on a maximum coverage model for summarization. Generally speaking, their model assumes that documents contain several concepts which link the ideas in the document. Ideally, a concept is a logical and independent fact, such as those used in the PYRAMID method [NEN 04]. This requires manual identification, and they opted for concepts based on n-grams, which are estimated by their frequency in the input *corpus*. However, [GIL 08] admit that the use of more sophisticated concepts would have been preferable. A "concept" was defined as a bigram of stems. Maximizing these concepts was inspired by recovery in the ROUGE-2 metric [LIN 04]. Words included in the antidictionary are ignored; however, bigrams containing at least one relevant word are preserved. Sentences which have no word in common with the query are ignored. To evaluate the relevance of a concept, authors use the frequency of the appearance of words which constitute this concept from the documents in the group to be summarized.

Reference	Summary A	Summary B
The European Airbus A380 flew its maiden test flight from France 10 years after design development started. The A380 super-jumbo passenger jet surpasses the Boeing 747 and breaks their monopoly. Airlines worldwide have placed orders but airports may need modification to accommodate the weight and width of the A380. U.S. airlines have not placed an order. Airbus has fallen behind in production and a backlog of orders has developed. Airbus must sell at least 250 planes to break even financially. The A380 is overweight and modifications to meet the weight requirements impacted the budget. Additional test flights are planned.	*The superjumbo Airbus A380, the world's largest commercial airliner, took off Wednesday into cloudy skies over southwestern France for its second test flight. The European aircraft maker, based in the French city of Toulouse, said the second flight – which came exactly a week after the A380's highly anticipated maiden voyage – would last about four hours. As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly.*	*The Airbus A380 has passed its emergency evacuation test, the European aircraft maker announced Wednesday, an important step toward the superjumbo's official certification for commercial flights. Both the European Aviation and Safety Agency and the U.S. Federal Aviation Administration approved the A380's performance in an emergency evacuation drill conducted in a darkened aircraft hangar Sunday with 853 volunteers and 20 on-board staff. One volunteer broke a leg and about 30 others sustained minor injuries during the drill in Hamburg, Germany, when all 873 participants evacuated the plane in under 80 seconds, using only eight of its 16 exits.*

Table 4.5. *Example of a reference summary and two automatic summaries for the update summarization task (TAC '08)*

A relevant concept is defined as a bigram which appears in more than 40% of documents. The algorithm considers this optimization as a problem of integer linear programming (ILP). The function to be maximized is the number of different relevant concepts which appear in the summary:

$$\begin{aligned} &\max \quad \sum_i w_i c_i \\ &\text{subjet to} \sum_j l_j s_j < L \end{aligned} \qquad [4.13]$$

where w_i is the weight of the concept i and c_i is a binary variable which indicates the presence of the concept in the summary; s_j is a binary variable which indicates the presence of the sentence j in the summary, l_j is the sentence length j and L the maximum length of the summary.

The constraints of the function [4.13] are as follows:

– the number of words in the summary (100);

– the fact that a sentence is selected if and only if all its concepts are also selected;

– the fact that concepts and sentences have the values of 0 or 1 to indicate the presence or the absence of the concept and the sentence in the summary.

The number of occurrences of a concept in the summary is ignored which should reduce redundancy while increasing informative coverage. The summary which maximizes the function [4.13] is selected. Sentences are sorted in chronological order in the source articles.

In preprocessing, sentence segmentation is realized with the NLTK PUN-KT segmenter[14] [KIS 06], which gives comparable results to some of the best systems based on rules and supervised. In postprocessing, the ICSI system ordered the selected sentences in the most simple way. Sentences are firstly sorted according to the date of

[14] See Appendix A.1.3.

the source document, then by their order in the source documents. The main advantages of this approach are that it does not require any knowledge and it remains quite independent from language. The 2009 version of this system International Computer Science Institute, Berkeley/University of Texas, Dallas (ICSI/UTD) [GIL 09] made linguistic improvements and took the position of sentences in the documents into account. Table 4.6 shows the performance of the ICSI system (best results) for the main task (set *A* and *B*) at TAC 2008 and Table 4.7 shows its performance (system ICSI/UTD with compression) for the main task at TAC 2009.

Evaluation	Set A		Set B	
	Score	Rank	Score	Rank
Responsiveness	2.689	9	2.167	22
Linguistic quality	2.479	21	2.479	24
PYRAMID	0.345	2	0.255	16
ROUGE-2	0.111	1	0.088	11
ROUGE-SU4	0.143	1	0.125	13
BASIC ELEMENTS (BE)	0.064	1	0.059	14

Table 4.6. *Results of the ICSI system (set A to the left and set B to the right) for the main task at TAC '08*

Evaluation	Set A		Set B	
	Score	Rank	Score	Rank
Responsiveness	5.159	1	4.750	2
Linguistic quality	5.636	5	5.523	6
PYRAMID	0.383	1	0.304	2
ROUGE-2	0.122	1	0.104	1
BASIC ELEMENTS (BE)	0.064	1	0.062	2

Table 4.7. *Results of the ICSI/UTD system (set A to the left and B to the right) for the main task at TAC '09*

4.6.5. *The CBSEAS system at TAC*

[BOS 09]'s work presents a multidocument text summarization system which took part in the TAC campaigns in 2008 and 2009. It is composed of the following stages:

– *Preprocessing*: the input documents are cleaned, orthographically normalized and segmented into sentences.

– *Sentence representation*: the following elements are added to the sentences:

- parts-of-speech (POS) morphosyntactic tagging with TREETAGGER;
- named entity (EN) recognition;
- identifying terms;
- calculating the centroid with MEAD, a method described by [RAD 00] (see section 4.5.5).

– *Sentence similarity*: the similarity between sentences is calculated using the Jaccard index.

– *Semantic classification*: sentences are semantically classified using the *Fast global k-means* algorithm.

– *Sentence selection*: the most relevant sentences, according to a double centrality (local in its semantic group and global in relation to the centroid or the user's query), are extracted to constitute a summary.

– *Postprocessing*: sentences at the top of the structure of the semantic graph are displayed first.

Figure 4.13 shows the general architecture of the CBSEAS system.

At the TAC 2009 campaign, the CBSEAS system obtained very good results for content: 9 out of 53 in the ROUGE evaluation and 11 out of 53 in PYRAMIDS. In relation to readability, the results were not as good: 31 out of 53, leaving room for improvement [POI 11]. CBSEAS was also used at the TAC 2008–2009 opinion summarization task. For this purpose, CBSEAS was modified to take the polarity of opinions into account. This modification will be presented in section 6.5.1.

4.7. Multi-document summarization by polytopes

A recent method of extractive text summarization consists of considering MDS as an optimization problem. To this end, the VSM is extended to a hyperplane model. [LIT 13] represent a document as the sentence–term matrix S whose inputs contain values of the occurrence of words and display the matrix as a set of intersecting hyperplanes. The potential summaries are represented as the intersection of two or more hyperplanes. An additional constraint is used to limit the number of words used in a summary. A summary is correct if the frequency of words is preserved during the summarization process and, in this event, the summarization problem is reduced to the problem of finding the point on a convex polytope (defined by linear inequalities) which is closest to the hyperplane describing all the word frequency in the document. The method known as POLY^2 has a polynomial order complexity, i.e. the time required to resolve the problems in linear programming.

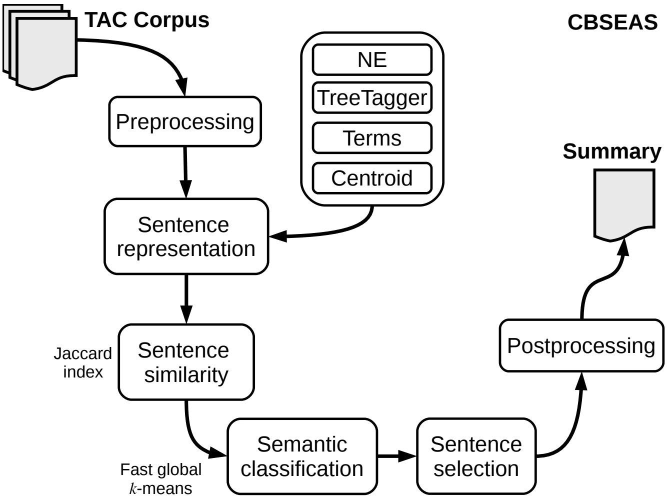


Figure 4.13. *Architecture of the multidocument summarization system CBSEAS*

[LIT 12, LIT 13]'s works gave interesting performances in MDS and multilingual summarization. At the Association for Computational Linguistics (ACL) Multiling 2013 pilot task (see section 5.2), the POLY^2 system obtained the following ranks:

– English: 7/10 in all ROUGE metrics (stemmed);

– Hebrew: 3/9 in ROUGE-1, 5/9 in ROUGE-2, 4/9 in ROUGE-3 and ROUGE-4;

– Arabic: 7/10 in ROUGE-1,2, 6/9 in ROUGE-3, and 5/10 in ROUGE-4.

4.8. Redundancy

Redundancy presents a more serious problem to MDS systems than to single-document summarization systems. Indeed, in MDS, the source documents are generally written by different authors, although they concern the same topic.

Let us consider, for example, the sentences in Table 4.8, which were extracted by an algorithm from a multidocument *corpus*.

At the level of redundant information, it is clear that some of these sentences must be eliminated or at least fused together as they say more or less the same thing (particularly sentences 1 and 6). But, how and at what moment should the phenomenon of redundancy in MDS be eliminated or reduced? Before extraction or before the generation of the summary?

Sentence	The invention of sushi
1	Raw fish sushi was invented in Japan in the 16th Century.
2	Fish sushi was invented by the inhabitants of Japan.
3	In Japan, a new fish-based dish, was invented (two centuries ago).
4	Sushi, a Japanese invention, in 1500.
5	The creation of sushi by the Japanese in the 1500s.
6	Luxury dish in Europe, sushi was an invention of Japanese fishermen in the 16th Century.

Table 4.8. *Example of redundant sentences extracted from a multidocument corpus*

Several strategies have been introduced to manage redundancy in MDS:

– calculating cosine similarity [VAN 79] between sentences which are already part of the summary and a new candidate sentence;

– calculating MMR [CAR 98] or SMMR [BOU 09a] to measure a compromise between diversity and similarity;

– calculating the informativity of sentences by the measure χ^2 [MAN 99b] [15]. This measure enables candidate sentences which are too semantically close to sentences in the summary to be identified;

– calculating the temporality of documents from temporal tags [MAN 00];

– calculating informativity from documents' meta-information [SWA 00], etc.

[NEW 04]'s work comparing several antiredundancy techniques shows that a cosine similarity measure between sentences based on a simple zero-knowledge vector [VAN 79] gave a similar performance to other more complex methods, such as latent semantic analysis (LSA) [DEE 90]. Other research into reducing redundancy focused on the temporality of documents. For instance, a method for generating update summaries [16] consists of extracting temporal tags [MAN 00] (dates, periods which have passed, temporal expressions, etc.) or constructing a chronology from the documents' meta-information, if available [SWA 00].

Another way of reducing redundancy consists of fusing semantically close sentences. An unsupervised sentence fusion method[FIL 10] was developed to generate short and linguistically simple sentences, while avoiding redundancy in the final summary. Multisentence fusion is a step toward paraphrasing, one of the stages required in the generation of abstracts. This technique will be described in section 7.6.

4.9. Conclusion

MDS is a much more complex task than single-document summarization. In addition to having a *corpus* of several documents

[15] "A chi-squared test, also referred to as chi-square test or χ^2 test, is any statistical hypothesis test in which the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true" (source: Wikipedia: http://en.wikipedia.org/wiki/Chi-squared_test).

[16] Update summarization for the identification of new information is studied in section 4.6.

about a topic, the user's information needs must also be taken into account. This information need is expressed as a query which guides the production of multidocument summaries. The phenomenon of redundancy presents systems with additional difficulties. Indeed, sentences from several documents concerning the same topic can contain repetitions. Therefore, systems must manage redundancy properly so that the readability and the informativeness of the summary produced can be guaranteed. Several antiredundancy methods exist (MMR, temporal tags, similarity during the construction of the graph, etc.). A method of fusing semantically close sentences with an incremental graph has also been presented.

Creating update summaries is a much more difficult task than creating the main summary for automatic systems. The DUC/TAC evaluations showed that in this task systems obtained low scores for responsiveness and in PYRAMID. Moreover, the gap between automatic summaries and those produced by people has grown. This is reflected in the scores for overall responsiveness, overall readability and in PYRAMID. Finally, the NIST baseline was the best in terms of readability, but had very low PYRAMID information content. One of the major difficulties is evaluating and optimizing the number of words representing "new" information. If too many words are extracted, the summaries produced will have little in common with the topic. Conversely, if too few words are retained, the readability of summaries will be reduced due to high redundancy. Experiments show that the addition of new information impacts positively on the quality of summaries: the information is divided more heterogeneously, syntactic redundancy is reduced and readability as well as general quality improve. The systems presented in this chapter use simple approaches that avoid redundancy. They select sentences which are closely related to the topic and ignore information that is already known. These systems give very interesting performances.

PART 2

Emerging systems

5

Multi and Cross-lingual Summarization

The large number of digital documents available on the Internet in several languages has prompted automatic summarization to move toward the problem of multilingualism. Multilingual systems have been developed from the perspective of multidocument summarization. One of the first multilingual systems was Summarist, created in 1998 by Edward Hovy and Chin-Yew Lin. Columbia NewsBlaster, an algorithm from the University of Columbia, is a multilingual multidocument system that has been producing summaries online since 2001. Compare and contrast program for summarization (Caps) is an English-Arabic multidocument system that produces summaries in English, Arabic or both. Mead, originally a bilingual summarizer, has become multilingual over time. NEWSEXPLORER, NewsBrief and Google News are examples of online multilingual systems. The task of automatic cross-lingual multidocument summarization involves generating a summary in a different language to that of the source documents. We will present an approach to cross-lingual multidocument summarization that takes into account the scores related to the translation quality during the sentence weighting process.

5.1. Multilingualism, the web and automatic summarization

According to the briefing paper "Multilingualism and the Internet" from the InternetSociety.org (ISOC): *"Multilingualism is a fact of life in countries where many languages coexist within the same country or region or even in much smaller communities"*[1]. Unfortunately, multilingualism is not as well-established in the online world (the Internet) as in the real world. The Internet is no longer unilingual, like

[1] Report available from: http://www.internetsociety.org/sites/default/files/multilingualism-20090514.pdf

at the beginnings of the Web in the 1990s, however, very few of the some 7,000 languages of the world are used in a significant way. Table 5.1 shows the position of the top 10 languages used online in May 2011. Table 5.2 shows the top 10 languages used on Twitter in 2013. András Kornai's paper, published in October 2013 [2], shows worrying results with regard to the languages used online. In fact, less than 5% of the world's languages are in use on the Internet. English, the French, Italian, German, Spanish (FIGS) languages, the Chinese, Japanese, Korean (CJK) languages and the languages of former colonial empires (Dutch, Portuguese and Russian) come out on top. This study confirms the results shown in Tables 5.1 and 5.2. In Kornai words': *"Of the approximately 7,000 languages spoken today, some 2,500 are generally considered endangered. Here we argue that this consensus figure vastly underestimates the danger of digital language death, in that less than 5% of all languages can still ascend to the digital realm"*.

Internet users by language			
Rank	**Language**	**Millions**	**Percentage %**
1	English	565	■■■■■■■■■■■■■■■■■■■■■■■■■■■ 26.8%
2	Chinese	510	■■■■■■■■■■■■■■■■■■■■■■■■ 24.2%
3	Spanish	165	■■■■■■■■ 7.8 %
4	Japanese	99	■■■■■ 4.7%
5	Portuguese	83	■■■■ 3.9%
6	German	75	■■■■ 3.6%
7	Arabic	65	■■■ 3.3%
8	French	60	■■■ 3.0%
9	Russian	59	■■■ 3.0%
10	Korean	39	■■ 2.0%
Top 10		1,616	82.2%
Others		351	■■■■■■■■■■■■■■■■■■ 17.8%

Table 5.1. *The top 10 languages used on the web in May 2011 (source: http://www.internetworldstats.com/stats7.htm)*

Many digital resources are currently available in multilingual versions. For instance, the European Union's official documents are

[2] [KOR 13]; source: http://www.plosone.org/article/info%3Adoi%2F10.1371 %2Fjournal.pone.0077056

drafted in 24 languages[3]. Consequently, tasks, such as cross-lingual information retrieval, *emerge* naturally: ideally, a user would formulate a question in a language he or she speaks fluently, the query would be executed on a collection of documents written in several languages, and the system would present its response in the user's language. The phenomenon of multilingualism does not only arise in Europe: on the Web, the multiplication of the number of documents written in the top 10 used languages (Tables 5.1 and 5.2) has made it necessary to develop methods of multi and cross-lingual information retrieval and extraction. Users do not understand or have a poor understanding of the languages dominating the Internet. This prevents them from accessing large amounts of information expressed in these languages. Other Natural Language Processing (NLP) tasks – such as text mining, document categorization, automatic text summarization etc. – can be applied to multilingual documents if their techniques are adapted.

Languages on Twitter		
Rank	**Language**	**Percentage %**
1	English	■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ 34%
2	Japanese	■■■■■■■■■■■■■■■■ 16%
3	Spanish	■■■■■■■■■■■■ 12%
4	Malay	■■■■■■■■ 8%
5	Portuguese	■■■■■■ 6%
6	Arabic	■■■■■■ 6%
7	French	■■ 2%
8	Turkish	■■ 2%
9	Thai	■ 1%
10	Korean	■ 1%
Top 10		88%
Others		■■■■■■■■■■■■ 12%

Table 5.2. *Ten most popular languages on Twitter in 2013. The percentage is an average number of tweets per day (source: http://www.mediabistro.com/alltwitter/twitter-top-10-languages_b52776)*

[3] There are 24 official languages in the European Union: Bulgarian, French, Maltese, Croatian, German, Polish, Czech, Greek, Portuguese, Danish, Hungarian, Romanian, Dutch, Irish, Slovak, English, Italian, Slovene, Estonian, Latvian, Spanish, Finnish, Lithuanian and Swedish. Source: http://ec.europa.eu/languages/policy/language-policy/official_languages_en.htm.

5.2. Automatic multilingual summarization

As seen in the previous chapters, automatic monolingual summarization (single and multidocument) has been an extremely important subject of research for a long time. However, most research, unsurprisingly, concerned English [MAR 00b, TEU 02]. Since then, other research into automatic summarization has been conducted in relation to European languages other than English, such as French [BOU 09b, TOR 02], Spanish [CUN 05, MAT 03], Portuguese [SAL 01], Catalan [FUE 04] or Swedish[4]. The Document Understanding Conference (DUC) and Text Analysis Conference (TAC) evaluation campaigns (see section 8.6) have also defined a certain number of specific tasks concerning multilingual summarization. For instance, at the DUC 2004 task 2 was short multidocument summaries focused by Topic Detection and Tracking Evaluation (TDT) events and task 3 was very short cross-lingual single-document summaries.

Multilingual summarization is a relatively new field of research, which has been tackled by adapting existing extraction-based multidocument systems to process English documents. Given source documents written in several languages, automatic multilingual summarization consists of generating a summary in a target language that is one of the source languages. Two classic solutions to resolve this problem are:

– to machine translate the generated summaries;

– to translate the documents before the sentence extraction phase.

The first solution is generally preferred over the second, because translating the documents beforehand makes sentence weighting more risky. Indeed, errors made by the machine translation system can creep into the process of producing the summary. Figure 5.1 shows an outline of the two classic architectures of multilingual systems.

Some systems, such as SUMMARIST [HOV 98], extract sentences from documents in a variety of languages and then translate the

[4] http://people.dsv.su.se/hercules/papers/Textsumsummary.html.

resulting summary (Figure 5.1(a)). Other systems, such as NEWSBLASTER [BLA 04, MCK 02], carry out the translation before extracting the sentences (Figure 5.1(b)).

Lenci *et al.* [LEN 02] developed a multilingual system that uses several linguistic resources. Readability is already a major problem for extraction-based systems. Indeed, the output of a machine translation system generally contains errors that further reduce readability. This is more obvious for languages that are linguistically distant, for instance, English, Chinese and Arabic (among others).

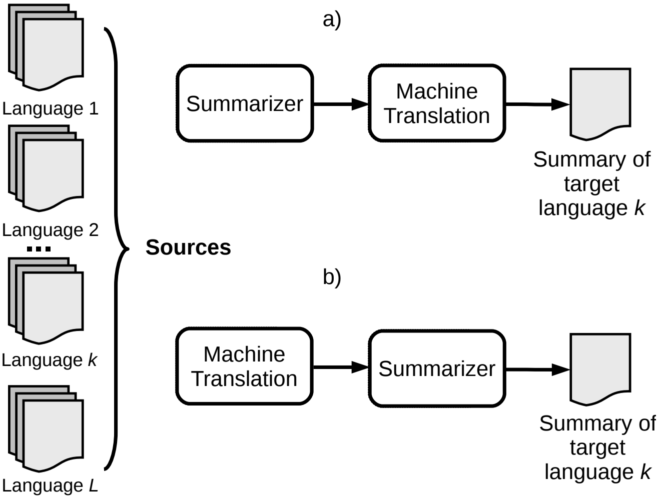


Figure 5.1. *Two classic approaches of automatic multilingual summarization systems: a) translation after the summary and b) translation before the summary*

At the international evaluation campaigns, new tasks were created to measure the performance of systems. At TAC 2011, the *Multiling Pilot* task was introduced, which sought to favor and promote the use of multilingual algorithms for automatic summarization. The Website for the TAC Multiling project[5] contains corpora of summaries

[5] http://multiling.iit.demokritos.gr.

available in the following languages: Arabic, Czech, English, French, Greek, Hebrew and Hindi.

In 2013, the Association for Computational Linguistics (ACL) Multilingual Multidocument Summarization Workshop introduced two main tasks and a pilot task (from the Multiling Website[6]):

– *Summarization task.* “This MultiLing task aims to evaluate the application of [...] language-independent summarization algorithms on a variety of languages. [...] in the MultiLing 2013 workshop [...] participating systems will be required to apply their methods on a minimum of two languages. Evaluation will favour systems that apply their methods in more languages. The MultiLing task requires to generate a single, fluent, representative summary from a set of documents describing an event sequence. The language of the document set will be within a given range of languages and all documents in a set share the same language. The output summary should be of the same language as its source documents. The output summary should be $\leq$ 250 words.”

– *Evaluation task.* “This task aims to examine how well automated systems can evaluate summaries from different languages. This task takes as input the summaries generated from automatic systems and humans in the summarization task [...] Ideally, we would want the automatic evaluation to maximally correlate to human judgement.”

– *Pilot: multilingual single-document summarization.* “This pilot aims to measure the ability of automated systems to apply single document summarization, in the context of Wikipedia texts. Given a single encyclopedic entry, [...] describing a specific subject, the systems will be requested to provide a summary covering the main points of the entry (similarly to the lead section of a Wikipedia page). The corpus will consist of (non-parallel) documents in over 40 languages. [...]”

In 2013, 10 languages were used for the multiling task: the seven languages used at the 2011 edition, in addition to Chinese, Spanish and Romanian. To evaluate the systems, the organizers brought in the following measures ROUGE (ROUGE-1, -2, -3 and -4) [LIN 04] using

[6] http://multiling.iit.demokritos.gr/pages/view/663/multiling-2013-tasks.

summaries provided by natives in each language as models and the AutoSummENG, MeMoG and NPowER systems [GIA 13].

In the following sections, several state-of-the-art multilingual multidocument systems as well as systems available online will be presented.

5.3. MEAD

MEAD [RAD 00, RAD 01a, RAD 03, RAD 04] was originally a system for the automatic generation of bilingual multidocument summaries. Since then, MEAD has become multilingual and can process documents written in English, Chinese, Dutch etc. MEAD implements several algorithms to weight and extract sentences based on the idea of a topic centroid. MEAD also introduced the notion of graph centrality. The LEXRANK algorithm was used to this effect (see section 3.5.4.1) as a measure for weighting sentences (centrality) in place of the centroid. The MEAD system was presented in detail in section 4.5.5.

5.4. SUMMARIST

SUMMARIST [HOV 98, HOV 99, LIN 98] is a multilingual multidocument system for extracting sentences. After extraction, it generates and translates the summary produced. SUMMARIST uses a topic identification, their interpretations and operations to generate text to produce a summary.

Figure 5.2 shows the general architecture of the SUMMARIST system. The system combines statistical techniques and characteristics of the *corpus*. The core of SUMMARIST is based on Lin's "equation" [LIN 98] :

$$\textbf{Summary} = \text{Topic identification} + \text{Topic interpretation}$$
$$+\text{Generation of summary}$$

To identify the important topics in the document (*topic identification*), SUMMARIST uses sentence position [BAX 58, EDM 69], keywords [EDM 69, PAI 90, TEU 97] and the occurrence of words. The discursive structure is also taken into account via rhetorical structure theory (RST) [MAN 88]. SUMMARIST does not use a parser or a grammar, so vocabulary books in English (90,000 words), Japanese (220,000 words), Spanish (45,000 words), Arabic (60,000 words), Indonesian (110,000 words) and Korean (110,000 words) were constructed for the machine translation part[7]. This makes SUMMARIST a true multilingual summarization system.

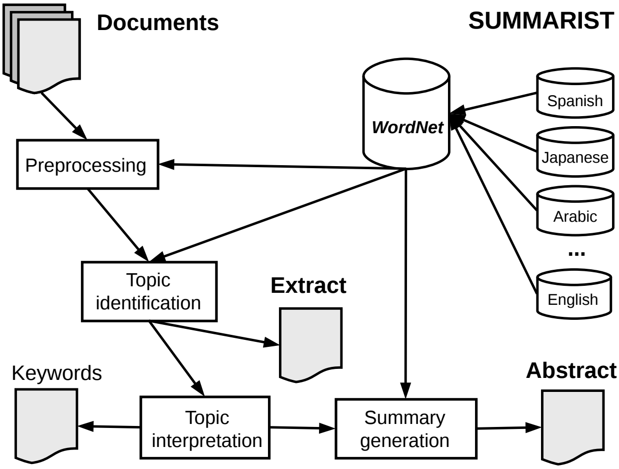


Figure 5.2. *General architecture of the automatic multilingual multidocument summarization system* SUMMARIST

Interpretation, the second step in the process of summarization, enables two or more topics to be fused in one concept. However, fusing topics such as { boy, menu, bill, food } into the concept “restaurant” is not evident. Simply counting the number of occurrences used in information retrieval is not enough to deduce the concepts. SUMMARIST uses the counting of concepts [LIN 95] and topic signature [LIN 97] to tackle the problem of concept fusion. Indeed, this

[7] www.isi.edu/natural-language/projects/SUMMARIST.html.

process has been considered as the most complex part of SUMMARIST, because it requires knowledge of the world that is not explicitly coded in the documents.

Generation is the final phase of the SUMMARIST summarization process. To do this, it calls on two modules used in natural language generation (NLG): the microplanner and the text realizer. In section 7.1, details about the process of generating the summary will be presented. The system is capable of producing extracts without generation, by displaying only the sentences retained during the *topic identification* phase. SUMMARIST also produces as output a list containing the keywords and concepts found. Finally, a sentence generator with a sentence weighting mechanism is used to produce abstracts.

This system has been applied to information retrieval in the MUST [LIN 99] system. MUST uses a translation of queries so that users can search for documents in a range of languages, then summarizes the documents with SUMMARIST and finally translates the generated summary.

5.5. COLUMBIA NEWSBLASTER

The automatic summarization system COLUMBIA NEWSBLASTER from the University of Columbia [EVA 01, MCK 01, MCK 02], enables multilingual and multidocument summaries to be generated from news taken from the Internet. On a daily basis and with no human intervention, COLUMBIA NEWSBLASTER produces summaries of documents that have been online from September 2001[8]. The

[8] Columbia Newsblaster is a system to automatically track the day's news. There are no human editors involved – everything you see on the main page is generated automatically, drawing on the sources listed on the left side of the screen. Every night, the system crawls a series of Websites, downloads articles, groups them together into "clusters" about the same topic and summarizes each cluster. The end result is a Web page that gives you a sense of what the major stories of the day are, so you do not have to visit the pages of dozens of publications (source: http://newsblaster.cs.columbia.edu/faq.html).

COLUMBIA NEWSBLASTER system comprises a rewriting module so that the generated summary has improved readability. References to people are rewritten to include their full name and a brief description the first time they are mentioned. Later in the text, people are referred to by their surname. In addition to improving readability, the rewritten version of the summary is generally shorter than the prerewritten version. Overly detailed descriptions are also eliminated. These changes are realized by comparing the sentence in the summary with the corresponding sentence in the original document [MCK 03]. Summaries are accompanied with images that have been recovered using precise patterns. Figure 5.3 shows the general architecture of the COLUMBIA NEWSBLASTER.

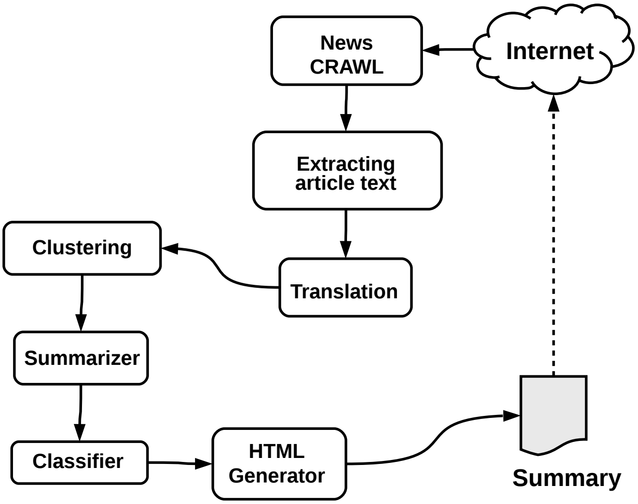
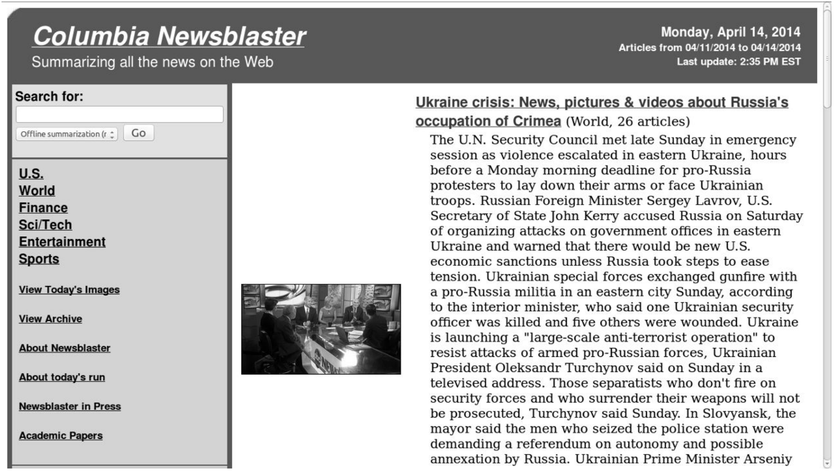


Figure 5.3. *Architecture of the automatic multilingual multidocument summarization system* COLUMBIA NEWSBLASTER

The COLUMBIA NEWSBLASTER can be consulted online at: http://newsblaster.cs.columbia.edu, unfortunately it is not available in free software. Figure 5.4 shows the system's user interface.

5.6. NewsExplorer

An online multilingual multidocument system is Europe Media Monitor (EMM) NewsExplorer from the European Union. NewsExplorer[9] automatically generates daily news summaries. The system applies a combination of Latent Semantic Analysis (LSA) (see section 3.4) and linguistic tools, as well as several multilingual technologies [KAB 13] about news articles automatically collected by EMM.



Monday, April 14, 2014
Articles from 04/11/2014 to 04/14/2014
Last update: 2:35 PM EST

Ukraine crisis: News, pictures & videos about Russia's occupation of Crimea (World, 26 articles)

The U.N. Security Council met late Sunday in emergency session as violence escalated in eastern Ukraine, hours before a Monday morning deadline for pro-Russia protesters to lay down their arms or face Ukrainian troops. Russian Foreign Minister Sergey Lavrov, U.S. Secretary of State John Kerry accused Russia on Saturday of organizing attacks on government offices in eastern Ukraine and warned that there would be new U.S. economic sanctions unless Russia took steps to ease tension. Ukrainian special forces exchanged gunfire with a pro-Russia militia in an eastern city Sunday, according to the interior minister, who said one Ukrainian security officer was killed and five others were wounded. Ukraine is launching a "large-scale anti-terrorist operation" to resist attacks of armed pro-Russian forces, Ukrainian President Oleksandr Turchynov said on Sunday in a televised address. Those separatists who don't fire on security forces and who surrender their weapons will not be prosecuted, Turchynov said Sunday. In Slovyansk, the mayor said the men who seized the police station were demanding a referendum on autonomy and possible annexation by Russia. Ukrainian Prime Minister Arseniy

Figure 5.4. *Interface of the* Columbia NewsBlaster *system, available for consultation online at http://newsblaster.cs.columbia.edu*

To produce summaries, the NewsExplorer system carries out the following tasks[10]:

– the day's news articles are recovered. This is done separately for each language, in clusters (groups) of connected articles;

– in each group, the system identifies named entities (names of people, places and organizations);

– variants of names used for the same person are identified;

[9] Source: http://emm.newsexplorer.eu.

[10] Source: http://emm.newsexplorer.eu/NewsExplorer/readme.html.

– monolingual groups are linked to similar groups in other languages;

– the most representative article in each group is identified, and its title is used for this group;

– extracted information is stored in a database. Each day, the system learns more about the people in this database;

– Wikipedia is consulted to recover and display images corresponding to the news and, in particular, to look for multilingual variants of named entities.

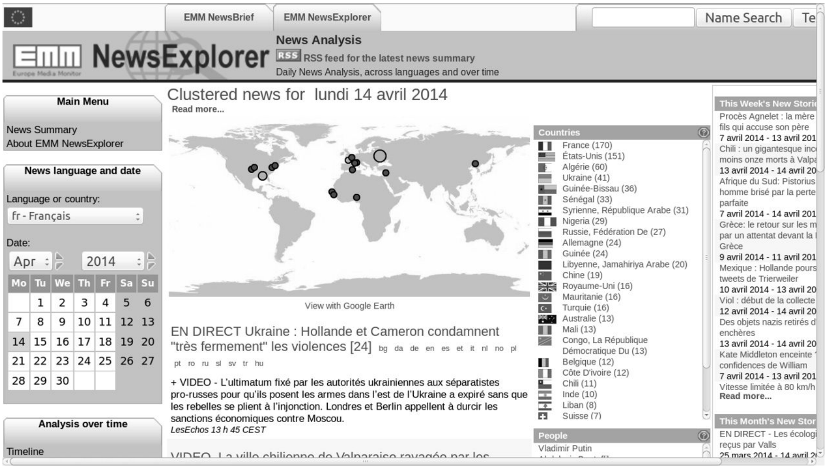
The basic tool for selecting the most important sentences is LSA. However, the LSA matrix has been improved by the multilingual identification and disambiguation of named entities. To do this, they used WordNet and the MeSH thesaurus [11], which (in the 2014 edition) contains 27,149 descriptors and approximately 218,000 entry terms with their hyponymic, hyperonymic etc. relations for the medical domain as well as for generic domains (animal names, food etc.).

Figure 5.5 shows the user interface for the EMM NEWSEXPLORER system.

An EMM system based on the previous one is the newswire aggregator NEWSBRIEF. The system presents an overview of newswires in the world that is updated every 10 min, 24/24 h and 7/7 days. The system is available in 28 languages (see Figure 5.6).

Kabadjov *et al.* [KAB 13] participated in the TAC 2009 campaign. Their system had access to multilingual tools for geotagging, entity disambiguation and also to the MeSH taxonomy. Their two submissions were numbered 11 and 19. The difference was that run 19 did not use the MeSH-based taxonomy. Both runs received very good scores for initial summaries of set A. Run 19 was the best overall run in linguistic quality, and run 11 was second. In TAC 2009, three different baselines were implemented (see section 4.6.2).

[11] MeSH is available from http://www.nlm.nih.gov/mesh/.



Language or country: fr - Français

Date: Apr 2014

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Clustered news for lundi 14 avril 2014

Read more...

EN DIRECT Ukraine : Hollande et Cameron condamnent "très fermement" les violences [24] bg da de en es et it nl no pl pt ro ru sl sv tr hu

+ VIDEO - L'ultimatum fixé par les autorités ukrainiennes aux séparatistes pro-russes pour qu'ils posent les armes dans l'est de l'Ukraine a expiré sans que les rebelles se plient à l'injonction. Londres et Berlin appellent à durcir les sanctions économiques contre Moscou.
LesEchos 13 h 45 CEST

VIDEO La ville chilienne de Valparaiso ravagée par les

Countries

France (170)
États-Unis (151)
Algérie (60)
Ukraine (41)
Guinée-Bissau (36)
Sénégal (33)
Syrienne, République Arabe (31)
Nigeria (29)
Russie, Fédération De (27)
Allemagne (24)
Guinée (24)
Libyenne, Jamahiriya Arabe (20)
Chine (19)
Royaume-Uni (16)
Mauritanie (16)
Turquie (16)
Australie (13)
Mali (13)
Congo, La République Démocratique Du (13)
Belgique (12)
Côte D'ivoire (12)
Chili (11)
Inde (10)
Liban (8)
Suisse (7)

People

Vladimir Putin

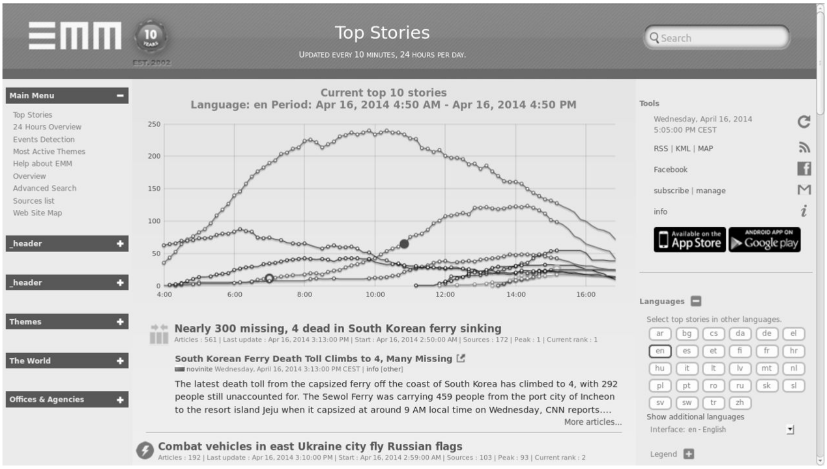
This Week's New Stori

Procès Agnelet : la mère fils qui accuse son père
7 avril 2014 - 13 avril 201
Chili : un gigantesque inc moins onze morts à Valpa
13 avril 2014 - 14 avril 20
Afrique du Sud: Pistorius homme brisé par la perte parfaite
7 avril 2014 - 14 avril 201
Grèce: le retour sur les m par un attentat devant la Grèce
9 avril 2014 - 11 avril 201
Mexique : Hollande pours tweets de Trierweiler
10 avril 2014 - 13 avril 20
Viol : début de la collecte
12 avril 2014 - 14 avril 20
Des objets nazis retirés d enchères
13 avril 2014 - 14 avril 20
Kate Middleton enceinte confidences de William
7 avril 2014 - 13 avril 201
Vitesse limitée à 80 km/h
Read more...

This Month's New Stor

EN DIRECT - Les écologi reçus par Valls
25 mars 2014 - 14 avril 20

Figure 5.5. *Interface for the* EMM NewsExplorer *interface, available for consultation online at http://emm.newsexplorer.eu*



Wednesday, April 16, 2014
5:05:00 PM CEST

Interface: en - English

Nearly 300 missing, 4 dead in South Korean ferry sinking
Articles : 561 | Last update : Apr 16, 2014 3:13:00 PM | Start : Apr 16, 2014 2:50:00 AM | Sources : 172 | Peak : 1 | Current rank : 1

South Korean Ferry Death Toll Climbs to 4, Many Missing
novinite Wednesday, April 16, 2014 3:13:00 PM CEST | info [other]

The latest death toll from the capsized ferry off the coast of South Korea has climbed to 4, with 292 people still unaccounted for. The Sewol Ferry was carrying 459 people from the port city of Incheon to the resort island Jeju when it capsized at around 9 AM local time on Wednesday, CNN reports....

Combat vehicles in east Ukraine city fly Russian flags
Articles : 192 | Last update : Apr 16, 2014 3:10:00 PM | Start : Apr 16, 2014 2:59:00 AM | Sources : 103 | Peak : 93 | Current rank : 2

Figure 5.6. EMM NewsBrief *system, available for consultation at http://emm.newsbrief.eu/NewsBrief/clusteredition/en/latest.html*

Table 5.3 shows the results (responsiveness, linguistic quality, Pyramid, Rouge and Basic Elements (BE)) of TAC'09 update summarization task.

ID system	Overall Responsiveness	Linguistic Quality	Pyramid score	ROUGE-2	ROUGE-SU4	BE
Best system ICSI	**5.159**	5.636	**0.383**	**0.121**	**0.151**	**0.064**
19	4.955	**5.932**	0.277	0.094	0.129	0.052
11	4.795	5.773	0.314	0.096	0.130	0.054
Baseline$_1$	3.636	5.477	0.175	0.063	0.099	0.029
Baseline$_2$	6.634	6.705	0.646	0.331	0.344	0.248
Baseline$_3$	6.341	7.477	0.358	0.106	0.138	0.053

Table 5.3. *TAC'09 results for the update summarization task (initial summaries of set A) obtained by Kabadjov et al. [KAB 13] and the three TAC baselines. The ICSI system was presented in section 4.6.4*

In the update summarization task, run 19 outperformed run 11. Run 19 was evaluated 9th in overall responsiveness, 14th in linguistic quality and 8th in PYRAMID. The fact that run 19 scored higher in all human-based scores (except for PYRAMID) with initial summaries suggests that taxonomic information has a positive impact on summary quality.

5.7. GOOGLE NEWS

The commercial system GOOGLE NEWS [12] automatically generates news in several languages with proprietary technology. According to its Website: "GOOGLE NEWS *is a computer-generated news site that aggregates headlines from news sources worldwide, groups similar stories together and displays them according to each reader's personalized interests"* [13]. In fact, GOOGLE NEWS offers this multidocument summarization system for most of the languages available on its translation site http://translate.google.com. Figure 5.7 shows the user interface of the GOOGLE NEWS system.

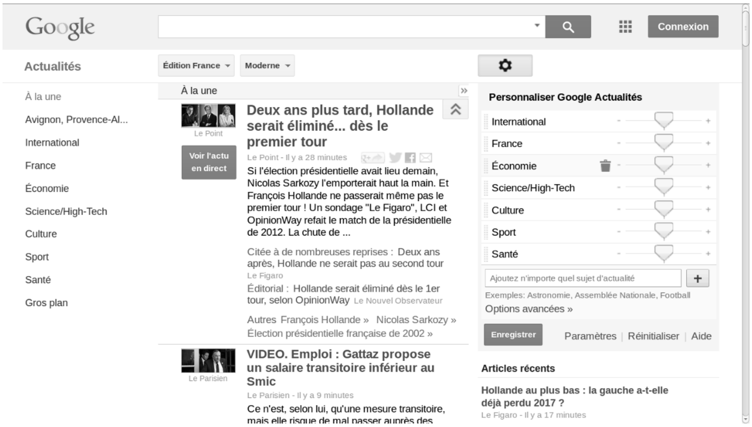
5.8. CAPS

[EVA 05] have researched the task of multidocument summarization in relation to bilingual journalistic bulletins. The

[12] Source: http://news.google.com.

[13] Source: http://news.google.com/intl/en_us/about_google_news.html.

system they developed, CAPS, is dedicated to produce summaries in English and Arabic. The approach followed by CAPS consists of summarizing information separately, only reports written in English, only reports written in Arabic or in a mixture of the two languages. The summaries generated by CAPS identify facts where the English and Arabic sources are in agreement as well as where there are explicit differences between the sources.



Deux ans plus tard, Hollande serait éliminé... dès le premier tour

Le Point - Il y a 28 minutes

Si l'élection présidentielle avait lieu demain, Nicolas Sarkozy l'emporterait haut la main. Et François Hollande ne passerait même pas le premier tour ! Un sondage "Le Figaro", LCI et OpinionWay refait le match de la présidentielle de 2012. La chute de ...

Citée à de nombreuses reprises : Deux ans après, Hollande ne serait pas au second tour Le Figaro

Éditorial : Hollande serait éliminé dès le 1er tour, selon OpinionWay Le Nouvel Observateur

Autres François Hollande » Nicolas Sarkozy » Élection présidentielle française de 2002 »

VIDEO. Emploi : Gattaz propose un salaire transitoire inférieur au Smic

Le Parisien - Il y a 9 minutes

Ce n'est, selon lui, qu'une mesure transitoire, mais elle risque de mal passer auprès des

Exemples: Astronomie, Assemblée Nationale, Football

Hollande au plus bas : la gauche a-t-elle déjà perdu 2017 ?

Le Figaro - Il y a 17 minutes

Figure 5.7. *Interface of the* GOOGLE NEWS *system, available for consultation online at http://news.google.com*

Acting as input for CAPS are two sets of documents about an event:

– a set of untranslated documents in Arabic and a set of documents in English;

– a set of manual or machine translations from Arabic documents and a set of documents in English.

In their experiments, Evans and Mckeown [EVA 05] used a machine translation system for documents written in English and Arabic and the SIMFINDER system [HAT 01] to calculate similarity. Information expressed exclusively in English and information expressed in both languages are summarized by selecting sentences from reports in English and have a certain fluency. However, summaries written in Arabic suffer from readability problems. Figure 5.8 shows the general architecture of the automatic multilingual summarization system CAPS.

5.9. Automatic cross-lingual summarization

Automatic cross-lingual summarization aims to produce a summary in a different target language to the language used in the source documents. As seen in section 5.2, strategies for implementing automatic multilingual summarizers consist of machine translation (before or after) the output of a classic automatic text summarization system. However, this approach is not without its disadvantages, since it is entirely dependent on the performance of machine translation systems.

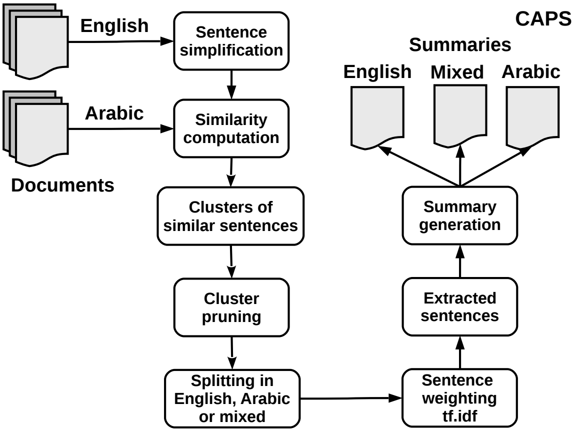


Figure 5.8. *Architecture of the automatic summarization system* CAPS *[EVA 05]*

It is more useful to try to include the notion of translation quality in the sentence weighting phase. This is the main idea of automatic cross-lingual summarization. [ORA 08] suggested using the maximal marginal relevance algorithm [CAR 98] (see section 4.5.2) to produce summaries of news written in Romanian and then machine translates them into English. [WAN 10] were interested in automatic single-document summarization from English into Chinese, using supervised methods to estimate the quality of the machine translation. Their research showed that taking the scores of translation quality into account enables both the content and readability of the generated summaries to be improved. The KEIZEI [OGD 99] system uses a translation of the query to enable Japanese and Korean users to search for documents in English and to produce summaries guided by a query, which targets segments containing terms from the query. Machine translation is a natural component of automatic cross-lingual text summarization systems. Unfortunately, despite considerable progress, machine translation systems still make errors that can have a strong negative impact on the quality of the summaries generated. Translation errors introduce incorrect information or make the generated sentences difficult to read. To reduce these effects, it is useful to take a score judging the translation quality into account so that incorrect translations are filtered during the summarization phase.

Among the different characteristics used to calculate quality are linguistic features that depend, or do not depend, on linguistic resources:

– syntactic analyzers (parsers) or WordNet;

– similarity measures between the source sentence and target sentence;

– internal characteristics of the translation system: the number of translations suggested by the source words or the sentence scores for the best translation hypotheses.

5.9.1. *The quality of machine translation*

Predicting translation quality has been considered a problem of binary classification [BLA 03] in order to distinguish good translations from bad translations. Other studies estimate a continuous value of the score either at the word level [RAY 09] or at the sentence level [RAY 09, SPE 09]. Several statistical classifiers estimate the quality of translations. *Corpora* of translations – automatically or manually tagged as correct or incorrect – have served to train these systems [QUI 04, SPE 09]. Automatic tagging was calculated using the Levenshtein distance [BLA 03], the BiLingual Evaluation Understudy Score (BLEU) score [PAP 02, RAY 09] or the National Institute of Standards and Technology (NIST) score [BLA 03, SPE 09], which are explained below.

5.9.1.1. *Levenshtein distance*

The Levenshtein distance measures the number of modification operations (in characters) required to transform a sentence (string) s_i into another s_j, in terms of the number of insertions, deletions and substitutions. The word error rate (WER) is derived from the Levenshtein distance, working at the word level instead of at the character level:

$$\text{WER}(s_i, s_j) = \frac{Ins(s_i, s_j) + Del(s_i, s_j) + Sub(s_i, s_j)}{|s_i|} \qquad [5.1]$$

where $Ins(\bullet)$ is the number of insertions, $Del(\bullet)$ is the number of deletions and $Sub(\bullet)$ is the number of substitutions carried out, in terms of words.

5.9.1.2. BLEU *score*

In this approach, the translation quality is between [0, 1]. It is measured by the statistical proximity to a series of reference translations. The BLEU formula counts the co-occurrences of n-grams between the output of the system s and the set of reference translations r_i, then the weighted geometric average is calculated. The formula is as follows:

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} \frac{1}{N} \log(P_n)\right) \qquad [5.2]$$

where N is the maximum n-gram size considered, and the precision n-gram P_n is calculated as follows:

$$P_n = \frac{\sum_{n\text{-grams}} \in \{s \cap r_i\}}{\sum_{n\text{-grams}} \in s} \qquad [5.3]$$

The numerator of the equation corresponds to the maximum number of co-occurrences of n-grams in s and in the reference translation r_i. The denominator is the total sum of the number of n-grams found exclusively in the candidate sentence s.

The brevity penalty BP (a value between [0, 1]) is used to penalize output sentences that are shorter than their reference sentences:

$$BP = \begin{cases} 1 & \text{if } |s| > |r_i| \\ \exp(1 - \frac{|r_i|}{|s|}) & \text{elsewhere} \end{cases} \qquad [5.4]$$

$|s|$ and $|r_i|$ are the length of the system output and the reference translation i, respectively.

5.9.1.3. NIST *score*

The NIST score [DOD 02] is similar to the BLEU score in that it also uses the precision of the co-occurrences of n-grams. However, it takes the arithmetic mean of the values of n-grams. Its formula is as follows:

$$\text{NIST} = BP \times \sum_{n=1}^{N} \frac{\sum_{\text{all the occurrences of } n\text{-grams}} \text{info}(n\text{-grams})}{\sum n\text{-grams} \in s} \qquad [5.5]$$

where info(n-grams) is calculated as follows:

$$\text{info}(n\text{-grams}) = \log_2 \frac{C((n\text{-}1)\text{-gram})}{C(n\text{-gram})} \qquad [5.6]$$

$C(n$-gram) is the count of occurrences of $(w_1, w_2, \ldots, w_n)$ and $C((n$-1)-gram) that of $(w_1, w_2, \ldots, w_{n-1})$ in the reference translations.

The penalty factor BP (also between [0, 1]), like in BLEU, penalizes translations that are too short in comparison to the average of reference translations $\langle|r_i|\rangle$. $\beta = 0.5$ is chosen to make brevity a penalty factor where the number of words in s is equal to 2/3 of the average number of words in the reference translations r_i. The equation to calculate BP is, therefore:

$$BP = \exp\left(\beta \times \log^2 \left[\min\left(\frac{|s|}{\langle|r_i|\rangle}, 1\right)\right]\right) \qquad [5.7]$$

5.9.2. *A graph-based cross-lingual English-French summarizer*

The GRAPH-SUM approach developed by Boudin *et al.* [BOU 11] summarizes and translates a set of source documents in English into French. This cross-lingual system uses a variant of TEXTRANK [14], a graph-based multidocument algorithm independent from language, and takes the quality of the machine translation in account during the sentence weighting phase. The idea is to minimize the impact of errors made by the machine translation system. Sentences selected to construct the summary will, therefore, have been judged to be both informative and easy-to-translate by the summarization system and by the machine translation system, respectively.

To do this, a supervised learning method by Support Vector Machines (SVM) was used to predict the scores of the translation quality. The scores were then integrated during the selection of informative sentences. The approach is realized in three stages. Sentences are machine translated and their translation quality is estimated. Sentences are evaluated according to their informativeness and translation quality score. Sentences with the highest scores are, therefore, extracted to constitute the summary.

Figure 5.9 shows an overview of the architecture of the cross-lingual multidocument system GRAPH-SUM.

5.9.2.1. *Predicting translation quality*

Source documents in English were split into sentences and then machine translated into French with Google Translate [15]. A machine translation score was calculated for each sentence in order to estimate the accuracy and fluency of the French translations [BOU 11]. Let s be the source sentence and s_t the target sentence. For each sentence, eight informative features about the difficulty of the translation and the readability of the generated translations were calculated:

– length in words $|s|$;

[14] See TEXTRANK in section 3.5.4.2.

[15] http://translate.google.com.

– the ratio $|s|/|s_t|$ of the lengths in words;

– the punctuation in s;

– the proportion of numbers and punctuation marks in s found in s_t;

– the perplexities[16] in s and in s_t (calculated with a language model[17] 5-grams LM);

– the perplexities calculated by reversing the word order in s and in s_t (reversed bigrams Language Model (LM)).

Out of the 84 features studied by Specia *et al.* [SPE 09], the first four were very relevant. Perplexity, aiming to estimate fluency, had already shown its efficiency in calculating the measure of confidence at the word level [RAY 09]. LMs are constructed from monolingual *corpora*[18].

Unlike other algorithms, the cross-lingual GRAPH-SUM method focuses on simple features that do not require linguistic resources such as parsers or dictionaries. Moreover, the translation scores are independent from the translator used. To predict translation quality, Boudin *et al.* [BOU 11] use *support vector regression* (ϵ-SVR), an extension of the SVMs used in the same application framework [RAY 09, WAN 10]. Calculations were realized with the A Library for Support Vector Machines (LIB-SVM) library [CHA 11] using Gaussian kernels. The NIST score had already been used for the same objective [BLA 03, SPE 09] and proved itself to have a greater correlation with human judgments than BLEU.

[16] "In information theory, perplexity is a measurement of how well a probability distribution or probability model predicts a sample. It may be used to compare probability models." (Source: Wikipedia: http://en.wikipedia.org/wiki/Perplexity).

[17] "A statistical language model assigns a probability to a sequence of m words $P(w_1, \ldots, w_m)$ by means of a probability distribution. Language modeling is used in many natural language processing applications such as speech recognition, machine translation, part-of-speech tagging, parsing and information retrieval". (Source: Wikipedia: http://en.wikipedia.org/wiki/Language_model)).

[18] Comprising 991 million words in English and 325 million words in French.

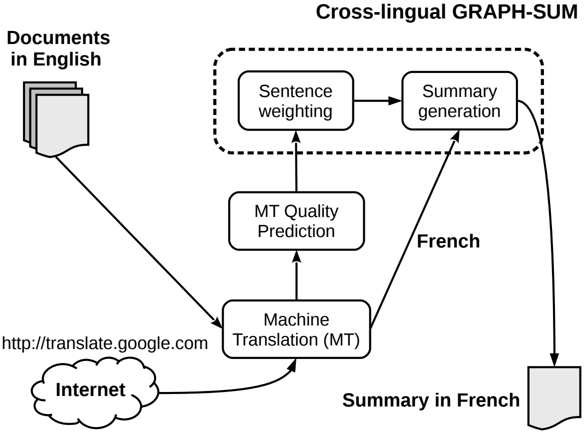


Figure 5.9. *Architecture of the cross-lingual multidocument summarization system* GRAPH-SUM

5.9.2.2. *Including translation quality scores*

GRAPH-SUM modifies the TEXTRANK sentence weighting function – based on a directed graph – to consider the cross-lingual aspect for this application. Intersentence similarity $w(\bullet)$ (equation [3.11]) was modified to include the machine translation quality score in equation [3.10]:

$$\omega(s_i) = (1-d) + d \sum_{s_j \in \text{In}(s_i)} \frac{w_{j,i} \times \text{prediction}(j)}{\sum_{s_k \in \text{Out}(s_j)} w_{jk}} \omega(s_j) \qquad [5.8]$$

where $\text{prediction}(j)$ is the machine translation quality score of the sentence j (see section 5.9.2.1). A fluent and correctly translated sentence sees its outgoing edges weights reinforced, and the sentence will consequently play a more central role in the graph. The sentence weighting algorithm has been modified to take advantage of the specificity of documents. Several studies [BRA 95, HOV 99, LIN 97] show that journalistic articles generally present a concise description of the subject at the beginning. Therefore, the weight of the outgoing arcs of the node corresponding to the first sentence has been doubled. Moreover, duplicate sentences and sentences containing less than five words have been eliminated.

To generate the summary, the algorithm follows a different MMR approach: during the construction of the graph, no arc is added between two nodes whose similarity exceeds a maximum threshold [MIH 05b]. To reduce redundancy, an additional step is included during the generation of summaries [GEN 09a]. All candidate summaries are generated from the combination of N sentences with the best scores, taking care that the total number of characters is optimal (i.e. below a given threshold and so that it is impossible to add another sentence without exceeding this threshold). The summary retained *in fine* is that with the highest overall score. This score is calculated as the product of the diversity of summary score – estimated by the number of different n-grams – and the sum of the sentence scores. Finally, to improve the readability of the summary, sentences are also sorted into the chronological order of the documents. If there are two sentences from the same document, the original order within the document is preserved. This maximizes temporal coherence.

An evaluation was realized on the DUC 2004 *corpus*: 50 sets of English documents. Each set deals with the same topic and contains on average 10 newspaper articles from the *Associated Press* or the *New York Times*. The task consists of generating a generic summary of less than 665 characters – including spaces and punctuation – from the multidocument set. A semi-automatic evaluation of the content and manual evaluations of the readability, carried out on a sample of documents, were realized.

The ROUGE-1, -2 and SU4 metrics were calculated. Four reference summaries in English were provided for each of the sets of documents. Three people translated the summaries available for the subset of 16 groups of documents chosen at random. Each summary was translated sentence by sentence, and no additional information was introduced[19]. The ROUGE evaluation was realized in a different framework to that of the usual tasks: producing summaries in French from sources in English. Several modifications were, therefore, effectuated. No strict

[19] Anaphora, disambiguation of proper nouns or reduction of redundancy information.

constraint was imposed on the size of summaries translated into French. Yet, the algorithm generated summaries whose total length of corresponding sentences in English respected the DUC 2004 constraint of 665 characters. The length of the reference summaries in French, therefore, increased 25% on average, in relation to the corresponding summaries in English.

Table 5.4 shows the ROUGE scores for different methods: the score of the best system during the DUC'04 campaign, the score obtained with GRAPH-SUM and the score of a baseline that took the first sentence of the most recent documents of each set to be translated. GRAPH-SUM performed well: its difference from the best system is not statistically significant.

System	ROUGE-1	Rank	ROUGE-2	Rank	ROUGE-SU4	Rank
Best system	0.3824	1	0.0922	1	0.1332	1
GRAPH-SUM	0.3805	2	0.0857	4	0.1311	3
Baseline	0.3238	26	0.0641	25	0.1029	29

Table 5.4. *Average* ROUGE *scores measured on DUC'04 data and ranks obtained in relation to the 35 participants of the campaign*

In cross-lingual evaluations, baseline summaries were obtained by translating the English summary produced by GRAPH-SUM. The ROUGE scores are presented in Table 5.5.

System	ROUGE-2	ROUGE-SU4	Readability
GRAPH-SUM baseline	0.1025	0.1371	2.06
GRAPH-SUM cross-lingual	**0.1069**	**0.1388**	**2.28**

Table 5.5. ROUGE *scores and average readability calculated on the DUC'04 subset translated into French*

Cross-lingual GRAPH-SUM had slightly better scores in ROUGE-2 and -SU4. However, this progress was not statistically significant. This can be explained by the fact that this method favors sentences with a good machine translation quality score. Therefore, sentences with weaker information content can be introduced into the summary, limiting improving the results. Linguistic quality was manually evaluated if the summaries were readable but equally comprehensible. Table 5.5 shows the average results from the evaluation obtained on the subset of 16 groups of documents. The judges stressed that this method improves the readability of the generated summaries. This shows the interest of using machine translation quality scores to improve the linguistic quality of summaries.

5.10. Conclusion

Multilingual multidocument summarization adds to the complexity of processing several source documents: it analyzes texts in different languages. SUMMARIST was one of the first multilingual and multidocument summarization systems. It combines statistical, discursive and knowledge interpretation methods to generate extracts and abstracts from documents written in several languages. Since 2001, the COLUMBIA NEWSBLASTER system has been producing summaries online from multilingual source documents present on the Internet. Classic systems combining machine translation and automatic summarization techniques have been created for multilingual summarization.

Cross-lingual summarization approaches aim to consider the quality of the machine translation in the sentence weighting process, linking translation and summarization systems more closely than in classic multilingual systems. Among these systems, an approach based on graph models for automatic cross-lingual and multidocument summarization has been presented. The machine translation quality scores were introduced at the construction of the graph stage. Sentence weighting according to popularity selects both the most informative and the most readable translated sentences. This method showed a marked improvement in readability, while the informativeness of summaries was maintained.

6

Source and Domain-Specific Summarization

From 1950–1970, research into automatic summarization essentially centered on documents with a general discourse, with two exceptions: Luhn's pioneering experiments conducted on technical and scientific corpora [LUH 58] and Pollock & Zamora's research into summarizing chemistry documents [POL 75]. In the 1990s, several researchers started to work on documents containing a specialized discourse, but as a general rule, the research used the same strategies as those used for summarizing documents with a general discourse. In this chapter, four automatic summarization systems for specialized documents will be presented. Yet Another CHemistry Summarizer (Yachs2) is dedicated to documents about organic chemistry; SummTerm is a system which uses semantic resources to generate summaries of medical articles. The third system is a summarizer combining statistical and linguistic algorithms, applied to the domain of biomedical articles. Finally, there is a summarization system for court decisions. With regard to source specific summarization, four systems oriented at summarizing online documents will be presented: web page summarization, microblog (tweet) summarization, email summarization and opinion summarization.

6.1. Genre, specialized documents and automatic summarization

A generic *corpus* is chiefly composed of newspaper articles often from the same period and even about the same topic, though without targeting any particular domain. This was the case, for instance, for the *corpora* in the TIPSTER, NII Testbeds and Community for Information access Research (NTCIR), TREC and Document Understanding Conferences (DUC)/Text Analysis Conferences (TAC) campaigns, etc. However, a specialized *corpus* is composed of documents from a specialized domain: very often from the scientific or technical domains. These documents use specialized language. For instance, the *corpora* composed of medical documents taken from the medical database MEDLINE [1]. Another type of *corpus* is constituted of encyclopedic documents, where the texts are about specific subjects, though they do not make use of too specialized language. A concrete example of this is the *corpora* from Wikipedia [2]. Finally, a literary *corpus* is constituted from works and documents from the literary domain [3]. The language used concerns imaginary worlds and situations or a poetic discourse. Literary *corpora* are very difficult to analyze from the perspective of automatic systems and fall beyond the scope of this work.

Classic works have built automatic summarization systems applied to scientific or technical *corpora* relating to chemistry [LUH 58, POL 75]. From the 1990s, several researchers, such as [LEH 95, MCK 95a, PAI 90, RIL 93, SAG 00a], among others, started to work on documents from a specialized domain, where texts are structured according to a specialized discourse. Several studies have been conducted about the summarization of texts specializing in medicine [AFA 05, CUN 07a, VIV 10a], organic chemistry [BOU 08b, BOU 08c] and court decisions [FAR 04a]. However, these works almost exclusively used the same strategies as those used for the summarization of documents from a general discourse: the text or discourse structure (sentences, cue-sentences and cue-words), sentence position, named entities , statistical techniques and machine learning, etc. [AFA 05] has pointed out that the automatic summarization of medical texts has become an important area of research because professionals in the domain need to process an ever-growing number of documents. Some relevant work relating to this domain has been conducted, for example, by [DAM 02, GAI 01, JOH 02, KAN 01, LEN 02]. But once again, the techniques they use are not specific to the medical domain but to generic documents. [CUN 08] developed DISICOSUM, a model for summarizing medical articles

[1] http://www.ncbi.nlm.nih.gov/pubmed.

[2] http://www.wikipedia.org.

[3] From sites such as http://www.cervantesvirtual.com.

written in Spanish, which considers various information specific to the medical domain: lexical units relevant to the domain, genre, textual information and discourse structure.

[CUN 09] carried out research which combined several summarization systems for specialized texts. The hybrid system combines statistical and linguistic techniques. An automatic text summarization system using semantic-based and frequency distribution was shown in [REE 07]. This system uses lexical chains[4] to identify salient sentences using concepts derived from domain-specific resources and a frequency-distribution method to reduce information redundancy. The two algorithms were combined to form a hybrid method. Lexical chains had previously been used in automatic summarization by [BAR 97] and [SIL 00]. The novelty of [REE 07]'s approach is that of linking concepts semantically using the *Unified Medical Language System (UMLS)*.

The UMLS provides a representation of knowledge in the biomedical domain[5]. This representation classifies concepts according to semantic types and the relations (hierarchical and non-hierarchical) between these types[6]. [REE 07] identify the concepts using the METAMAP UMLS[7], which maps nominal sentences that correspond to UMLS semantic types.

Working with specialized documents involves a precise usage of the notion "term". The concept of term in specialized documents is based on the *Communicative Theory of Terminology* [CAB 99]. The concept

[4] The sequence of words and the semantic-lexical relations (essentially identity and synonymy) between them.

[5] UMLS is made up of three knowledge components: the Metathesaurus of medical concepts, the Semantic Network and the Specialist Lexicon. See: http://www.openclinical.org/medTermUmls.html and Bodenreider O., *The UMLS: integrating biomedical terminology*. Nucleic Acids Research 32:267-270 (2004), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC308795/.

[6] See http://www.nlm.nih.gov

[7] MetaMap is a highly configurable program developed by Dr. A. Aronson at the National Library of Medicine (NLM) to map biomedical text to the UMLS Metathesaurus (source: http://metamap.nlm.nih.gov).

of term in specialized documents is based on the *Communicative Theory of Terminology*According to this theory:

DEFINITION 6.1.– *A term is a lexical unit (of one or several words) which designate a concept in a given specialized domain.*

Following this definition, terms have a designative function, though evidently each word in a language does not have this function.

A term detector and extractor is a system whose main task is to detect the words, in a specialized text, which have a designative function (see [CAB 01] for an overview of this topic). It is important to point out that a term can be ambiguous. This means that, sometimes, the same lexical unit can designate two or several concepts or *vice versa*. For instance, *mouse* in zoology or engineering and computer science; *adenoid* or *ganglion* in medicine. Moreover, a concept can be designated by two or several terms, such as *storage battery* and *accumulator* in electricity.

Working with articles from a specialized domain is interesting for two reasons. First, summaries written by their authors – specialists in the domain – are available, enabling abstracts to be compared with the automatic summaries. Second, titles contain a large number of terms, which can be exploited. However, there is one unavoidable difficulty when it comes to evaluating the summaries of specialized documents: manually evaluating the quality is relevant for generic documents, but it is very difficult to implement for specialized documents. Evaluators must also be specialists and finding enough people capable of evaluating this type of document is not easy. Judges must have a deep understanding of the information in the documents: it is not enough for the summaries to have linguistic quality and coherence; it is most important that the information in the summary be precise and relevant, given that it is aimed at specialists. This makes it even more difficult and costly to manually evaluate this type of document.

6.2. Automatic summarization and organic chemistry

In 2011, more than 30 million chemistry articles were viewed on http://www.cas.org/ and the number of articles published grew exponentially, worsening the problem of the overload of information in this domain. Automatic text summarization can help users to find relevant information in this mass of documents. However, this raises a question: is the use of classic statistical techniques for sentence extraction, presented throughout this book, still relevant for documents specializing in chemistry? The answer is yes and no. Yes, because these techniques avoid heavy processing by using large linguistic resources dependent (though often non-existent) on the domain. No, because the specificity of documents about organic chemistry can cause a fall in the performance of text summarization systems, if only classic methods of preprocessing and extraction (seen up until now) are used. However, classic tools such as syntactic taggers or parsers do not perform well, without a significant, costly and, for the majority of cases, manual phase of adaptation. The problems faced by these tools are due to the specificity of the domain [BOU 08a]: a specific and extremely broad vocabulary, long sentences containing a large number of quotations, formulas and names of chemical substances, cross references, etc.

One possible solution is to adapt statistical tools from automatic text summarization to make them as independent from the domain as possible, while maintaining their performance.

6.2.1. YACHS2

[BOU 08a] created the summarization system YACHS2 by adapting the tools of automatic summarization for generic documents to the specialized domain of organic chemistry. As a summarizer of chemistry documents written in English, YACHS2 differs from a summarization system for generic documents on two points:

1) molecule names (generally chemical entities) are specifically identified in an attempt to protect them before a standard preprocessing operation;

2) weighting is carried out using the occurrence of molecule names and the similarity between the title and the sentences at the character level.

For the latter similarity, the system uses a modified Jaro-Winkler measure. Figure 6.1 shows the processing chain of the summarization system for organic chemistry YACHS2.

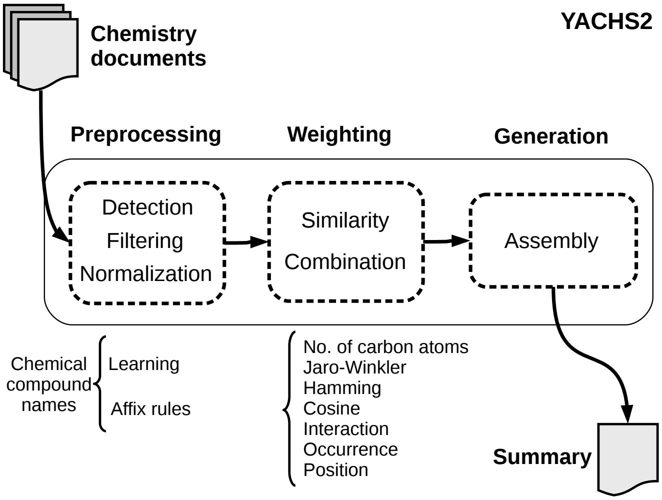


Figure 6.1. *Automatic summarization system for chemistry* YACHS2

6.2.1.1. *Preprocessing*

A standard preprocessing operation (splitting into tokens and sentences, eliminating punctuation, normalization of words by lemmatization or stemming) leads to significant information loss. Consider, for example, the following sentences:

– *7-Cycloalkynes are known to isomerize to the 1,2-α-dienes (CH3CH2)-CH2 under basic conditions*;

– *diagram of rotation about C2-C3 bond in butane (CH3CH2CH2)-CH3 per pair of eclipsed CH3-CH3 = 2.5 kcal/mol.*

Preprocessing can produce:

– *cycloalkyn know isomer dien under basic condit*;

– *diagram rotat about bond butan pair eclips kcal mol.*

The information about radicals and chemical molecules have been lost.

To overcome this difficulty, YACHS2 implemented a preprocessing operation specific to organic chemistry. Two algorithms have been developed:

– a Bayes probabilistic classifier based on the trigrams of letters (splitting into sliding window) where the probabilities were estimated from lists of words (chemistry and non-chemistry sets)[8];

– detection by patterns: writing regular expressions clustered into rules concerning the prefixes, infixes and suffixes of molecules.

The combination of detection algorithms produced a well-performing method. This strategy obtained a precision of 0.93 and an F-measure of 0.88, evaluated on a reference *corpus* (newspaper articles and their summaries, 20,000 words and 850 substance names) annotated by two people and validated by a specialist in the domain [BOU 08c].

6.2.1.2. *Sentence weighting*

For sentence weighting YACHS2 uses several classic measures (position, occurrences, etc.) and introduces two measures specific to the type of document in question:

– classic sentence weighting criteria:

- position of the sentence;
- quantity of informative words;
- cosine similarity between the sentences and the title (at the word level);
- degree of interaction (equation [3.18]);

[8] Learning was realized on a chemistry *corpus* of 10,000 substance names and a non-chemistry *corpus* of 198,000 words (*Spell Checking Oriented Word Lists* (SCOWL) *corpus*).

- sum of the Hamming distance (equation [3.24]);

– criteria specific to organic chemistry:

- quantity of molecule names and chemical entities;

- Jaro-Winkler similarity $\text{sim}_{JW}(T, s)$ between the title T and the sentences s (at the character level).

Jaro-Winkler similarity carries out a semantic correspondence between words in the same morphological family. For example, let the following title and sentence (after stemming and filtering) be [BOU 08a]:

titre $T =$ **cycloalky**lidenecarben provid ring expand **isom**

phrase s = **cycloalky**n know **isom**er dien under basic condit

They have no term in common; therefore, their cosine similarity is $\cos(T, s) = 0$.

However, their similarity at the character level is not null. Indeed, the Jaro-Winkler similarity $\text{sim}_{JW}(T, s) = 0.37$ due to the fact that s and T share two common subchains: **cycloalky** and **isom**.

The previous similarity criteria are combined in a decision algorithm (equation [3.33]) to weight the sentences in the *corpus*. The summary is generated by extracting the sentences with the highest weight. Molecule names and chemical entities are also signaled in the generated output. Figure 6.2 shows the user interface of the system YACHS2.

YACHS2 was evaluated [BOU 08c] following a protocol similar to that used during the National Institute of Standards and Technology (NIST) campaigns. The *corpus* is composed of 100 pairs of article summary from organic chemistry journals dating from 2000 to 2008: *Organic Letters, Accounts of Chemical Research* and *Journal of Organic Chemistry*, with a total of 8,923 sentences and 189,167 words.

The outputs of the YACHS2 system were compared to those of seven other systems: a random *baseline,* CORTEX, MEAD, OPEN

Text Summarizer (OTS)[9], Pertinence Summarizer with the use of linguistic markers specific to chemistry, Corpernic Summarizer and *Word*. Figure 6.3 shows the performance of Rouge-2 *versus* -SU4 for Yachs2. It clearly outperforms the other systems. Table 6.1 shows an output of an author summary compared with an automatic summary generated by Yachs2. It can be seen that the informative content of the author summary is present in the automatic summary: sentences 1a and 2a of the abstract are rewordings of the extracted sentences 1b and 3b. The author reused sentences in the document with a simple reformulation. The fluency and the coherence of the extract are also worth pointing out [BOU 08a].

Fichier Édition Affichage Historique Marque-pages Outils Aide
yachs2 (ver. 2.0.1)
http://daniel.iut.univ-metz.fr/yachs/
blaster
yachs2 (ver. 2.0.1)
Tutorial Video (new) | What's Inside ? | Contact
Welcome / Bienvenue to Yachs2
Upload a file | Copy & Paste
YachS2
LABORATOIRE INFORMATIQUE D'AVIGNON
Université d'Avignon et des Pays de Vaucluse
Copyright © 2008 Florian Boudin, 2009 Juan Manuel Torres
http://daniel.iut.univ-metz.fr/yachs/f_clipboard.php

Figure 6.2. *User interface of the automatic summarization system for organic chemistry* Yachs2, *available at: http://daniel.iut.univ-metz.fr/yachs*

[9] *The* OTS, http://libots.sourceforge.net, incorporates simple algorithms for normalization based on a lexicon of synonyms and a set of rules for stemming and analysis. Sentences are weighted by combining criteria concerning the presence of cue-words and frequency of words.

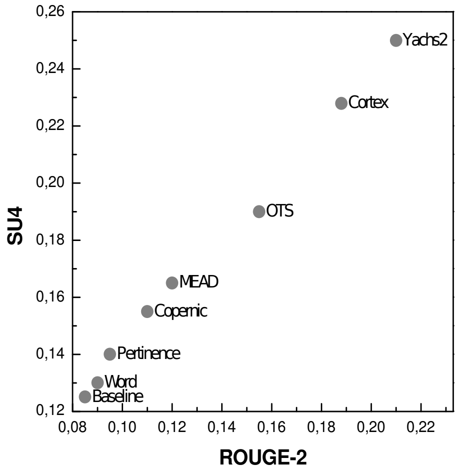


Figure 6.3. ROUGE *evaluation of* YACHS2 *versus several systems on a corpus about organic chemistry*

Sentence	Author abstract
1a	*A mild and general cycloaddition of arynes with iodonium ylides protocol has been developed for the synthesis of benzofurans.*
2a	*In the presence of CsF, ortho-silyl aryltriflates were reacted with iodonium ylides smoothly at room temperature in moderate to good yields.*
Sentence	Summary generated by YACHS2
1b	In the presence of *CsF, ortho-silyl aryltriflates* successfully underwent the *cycloaddition* reaction with *iodonium ylides* at room temperature to afford the corresponding *benzofurans* in moderate to good yields.
2b	Recently, much attention has been attracted to apply *ortho-silyl aryltriflates* as the *benzyne* precursors in organic synthesis because *ortho-silyl aryltriflates* could be readily converted into *benzynes in situ* under mild reaction conditions.
3b	In summary, we have developed a mild and general *cycloaddition* of *benzynes* with *iodonium ylides* protocol for the synthesis of *benzofurans*.

Table 6.1. *Example of* YACHS2*'s output versus an author summary (http://pubs.acs.org/doi/full/10.1021/ol800051k) [BOU 08a]*

6.3. Automatic summarization and biomedicine

The same automatic summarization problem facing organic chemistry is also facing medicine: there were 23,610,944 articles published on MEDLINE (*Medical Literature Analysis and Retrieval System Online*) on June 20, 2014.[10] Articles and quotations (approximately 15 million) have been indexed since 1966. MEDLINE has become doctors' and biologists' favorite research tool. However, this overload of information is counterproductive for information retrieval. Doctors require relevant information to be available and that this information be constantly updated; however, the public is increasingly asking for this specialized information to be made available to them. Author summaries are often too short and incomprehensible for most people. Automatic summarization can help give an overview of biomedical information to non-specialists.

However, the TAC 2014 summarization task aims to produce an automatic biomedical citance-focused summary. In TAC's words[11]: "Given: A set of Citing Papers (CPs) that all contain citations to a Reference Paper (RP). In each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP".

6.3.1. SUMMTERM

A feature of biomedical documents is that the title of articles is often very long and contains a large number of terms. For summarization algorithms, this can be a useful advantage. [VIV 10a] developed SUMMTERM, a semantic strategy for summarizing biomedical texts in Spanish. Inspired by Luhn's idea [LUH 58] that the terms in the title of a scientific text are relevant markers indicating the topic [BAR 97, REE 07, SIL 00], their hypothesis is that terms which are semantically linked to the terms in the title of a biomedical document are particularly relevant.

[10] Sources: http://ipubmed.ics.uci.edu/ and http://www.ncbi.nlm.nih.gov/pubmed.

[11] See: http://www.nist.gov/tac/2014/BiomedSumm/index.html.

The SUMMTERM automatic summarization algorithm improves sentences containing the term(s) in the title, as well as sentences containing terms which are semantically linked to the terms in the title. SUMMTERM uses two semantic resources: the ontology *EuroWordNet (EWN)* and the term extractor Yet Another Term Extractor (YATE).

The lexical databases WordNet and EuroWordNet

Developed at Princeton University [12], *WordNet* is a lexical-semantic ontology (network) for the English language [FEL 98, MIL 95]. It clusters words and their synonyms in a hierarchical structure. The basic semantic unit is the *synset* [13] which clusters several words which could be considered synonyms in certain contexts.

Lexical entries can be attached to several *synsets* because of polysemy. Most *synsets* are interconnected by a certain number of semantic relations which vary according to the type of words:

– nouns: hypernyms (X is a kind of Y), hyponyms (Y is a kind of X), holonym (X is part of Y) and meronym (Y is part of X);

– verbs: hypernyms (an action of X is a kind of action of Y), hyponyms (an action of Y is a kind of action of X), implication (if an action of X is realized, the action of Y is also realized);

– adjectives: associated nouns and the participle of the verb;

– adverbs: stem adjectives.

Nouns and verbs are organized into hierarchies, defined by their hypernymic relations. Synonymy sets have a unique index for sharing their properties. EWN is a multilingual extension of *WordNet*. The EWN version used comprises approximately 6,000 *synsets* (corresponding to approximately 10,000 variants) [VIV 10a].

[12] http://www.illc.uva.nl/EuroWordNet/.

[13] *Synset* is an acronym of synonymy set.

Nevertheless, other specialized data bases, such as *Systematized NOmenclature of MEDicine, Clinical Terms (SNOMED-CT)* include one hundred times more terms.[14]

*Yet Another Term Extractor (*YATE*)*

YATE is a tool for term detection and term extraction (TE) [VIV 01]. It combines several extraction techniques with the semantic knowledge in *EWN*. YATE is now available at http://iula05v.upf.edu/cgi-bin/Yate-on-the-Web/yotwMain.pl.

Given that the techniques depend on the different properties of candidate terms (CT), YATE combines a set of CT analyzers, which are briefly described below:

– domain coefficient: it uses the EWN ontology to sort the CTs;

– context: it evaluates each candidate with other candidates which are used in the same context as the sentence;

– classic forms: it tries to decompose lexical units in their format, taking into account form features of the countless terms in the domain;

– collocation method: it evaluates multiterm candidates according to their mutual information;[15]

– other internal and external CT analyzers.

The results obtained by this set of heterogeneous methods are combined using two strategies: a vote algorithm and a boosting algorithm. YATE uses ADABOOST, the most commonly used boosting implementation, and EWN as a source for specific information about the domain. EWN has enough coverage in the medical domain to develop a Spanish term extractor and use their relations in a semantic

[14] SNOMED-CT, a structured set of medical terms, broadly covers the clinical domain. Terms are organized into a certain number of hierarchies where the nodes are connected simultaneously by specialized vertical and horizontal relations. From 2002, a Spanish version of this resource has been regularly published on http://www.ihtsdo.org/.

[15] "In probability theory and information theory, the mutual information or (formerly) transinformation of two random variables is a measure of the variables' mutual dependence." (Source: Wikipedia: http://en.wikipedia.org/wiki/Mutual_information).

summarization algorithm. YATE was essentially developed for medicine, but it has also successfully been adapted to genetics and economics [JOA 08]. Figure 6.4 shows the general architecture of YATE.

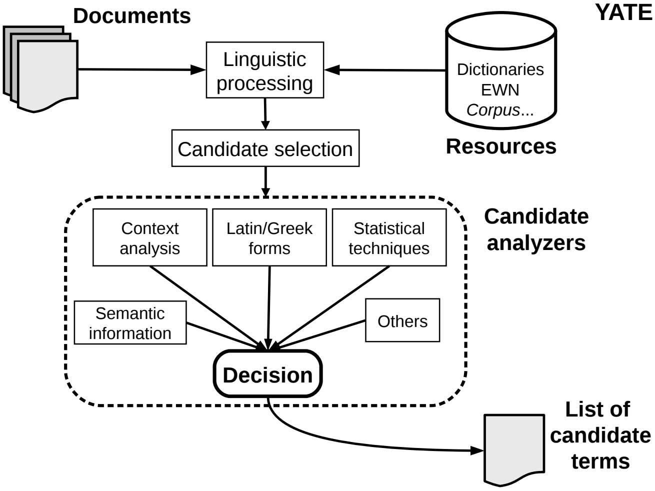


Figure 6.4. *General architecture of the term extractor* YATE *(adapted from [VIV 10a])*

YATE produces a list of "candidate terms", which are sorted according to their termhood.[16] For each term in the corpus, YATE measures the semantic distance between this term t and all the terms found in the title T. t receives a score(t) which is calculated using equation [6.1]:

$$\text{score}(t) = \delta \cdot \text{term}(t) + (1 - \delta) \cdot \max \text{sim}(t, t_T) \qquad [6.1]$$

where term(t) is the termhood of the term t, t_T is a term in the title of the document and δ is a weighting coefficient set at $\delta = \frac{1}{2}$ by default. The default value is a compromise between termhood and similarity.

[16] Termhood has been defined as "...the degree that a linguistic unit is related to domain-specific concepts" [KAG 96].

Sentence weighting

SUMMTERM weights the sentences in the document by considering both the termhood of words and the similarity of sentences whose terms are present in the title.

Two configuration parameters are required during input: the weighting coefficient δ and the compression rate τ, wanted for the final summary. A standard preprocessing operation (see section 2.1), shared by most Natural Language Processing (NLP) applications, is applied to the text: splitting into ρ sentences, tokenization and parts-of-speech (POS)-tagging. The weighting $\omega(s)$ of ρ sentences is realized in two stages: first, YATE is used to identify and give a score to the terms (equation [6.1]) present in the input *corpus*. Then, the score(t) of the terms in the sentence s are added up:

$$\omega(s) = \sum_{t \in L_{CAT}(s)} \text{score}(t) \qquad [6.2]$$

where $L_{CAT}(s)$ is the list of terms present in s. The final summary is generated by concatenating the sentences with the highest scores, while respecting the limit imposed by the compression rate. Figure 6.5 presents the general architecture of the semantic summarization system SUMMTERM.

To evaluate SUMMTERM, a *corpus* composed of 50 medical texts from the journal *Medicina Clínica*[17] was constructed. Each document contains four sections: “Introduction”, “Patients and methods”, “Results” and “Discussion”. Summaries containing 11 sentences (close to the length of the author abstracts) were generated. The SUMMTERM results were evaluated exclusively by ROUGE.[18]

[17] http://zl.elsevier.es/es/revista/medicina-clinica-2.

[18] As explained in Chapter 8, ROUGE is one of the most common systems for evaluating summaries. ROUGE offers a set of statistics which compare the candidate summaries to the reference summaries (summaries realized by people).

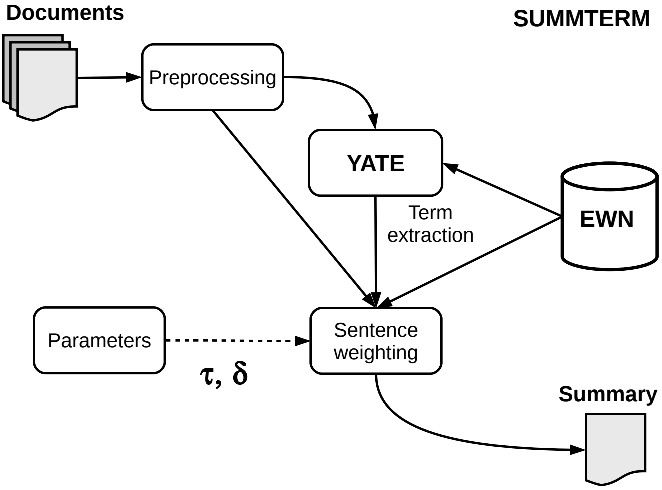


Figure 6.5. *Architecture of the semantic summarization system* SUMMTERM

The following systems were used to compare the performance of SUMMTERM: CORTEX, ENERTEX, PERTINENCE SUMMARIZER [19], OTS, SWESUM, *Word* and Lead sentences and random sentence baselines.

The authors' abstracts of the articles were the references. SUMMTERM was the fourth best performing system (ROUGE-2 = 0.333; SU4 = 0.385) after CORTEX, ENERTEX and OTS (see Figure 6.3.1). The scores of these four systems were relatively similar and they performed a great deal better than the other systems. Direct reading showed that the summaries produced by SUMMTERM include information from each of the four sections in the articles. This indicates that the algorithm maintains the coherence of the original text as a whole and not only certain specific sections. However, SUMMTERM seems to perform less well than other statistical methods, according to the ROUGE evaluations. A detailed analysis of the results showed that YATE was too conservative in its extraction of terms. This

[19] PERTINENCE SUMMARIZER was replaced by ESSENTIAL SUMMARIZER, see Appendix 2.

is particularly true for multiterms. Moreover, the list of CTs shows that the terms in specialized documents are frequently complex sequences of nouns with several adjectives. Furthermore, the terms in the EWN are not recognized as coming from their Latin forms due to their complexity.

SUMMTERM was also used to process biomedical texts written in Catalan. Fifty or so documents from a technical *corpus* from the IULA[20] were used. Given that this *corpus* does not contain any author summaries, evaluation with ROUGE was impossible. The automatic method FRESA [SAG 10] (see section 8.9.2) was, therefore, used to evaluate summaries. [VIV 10b] report average FRESA values of 0.862 for SUMMTERM, 0.841 for the OTS and just 0.815 for the Lead sentences baseline.

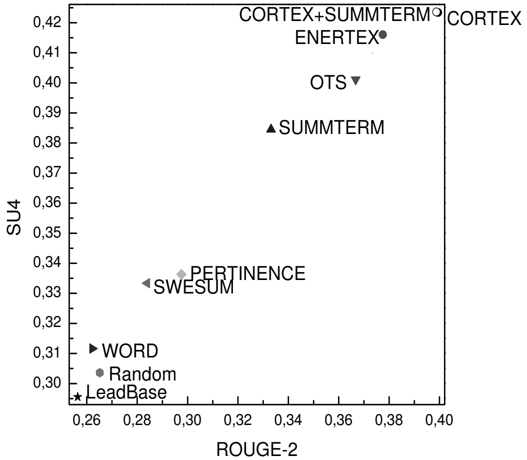


Figure 6.6. ROUGE-2 *versus* -SU4 *evaluation of the* SUMMTERM *system on a biomedical corpus*

[20] Institut Universitari de Lingüística Aplicada, Unversitat Pompeu Fabra (Barcelona), http://www.iula.upf.edu.

6.3.2. *A linguistic-statistical approach*

The majority of algorithms described throughout this book use very few linguistic resources or none at all. Though very effective, the performance of statistical systems reaches a ceiling as they are often compared at the sentence level of granularity. Moreover, processing fine dependences, such as anaphoric references, are very difficult to implement. One possible solution to the problem of statistics-based automatic summarization systems is to combine them with linguistic approaches.

It has been seen that there are a large number of statistical [BAR 97, KUP 95, MIH 04b, MIH 05a, RAD 00, SIL 00, TEU 97, TOR 02] or linguistic [ALO 03, MAR 98, ONO 94, TEU 02] automatic summarization systems in existence; however, very few of these combine the two criteria [ALO 03, ARE 94, ARE 96, NOM 94]. The idea is that, to generate a good summary, the linguistic aspects of a text are required in addition to the advantages of statistical techniques.

[CUN 09] were interested in analyzing the techniques used in automatic text summarization for specialized domains, specifically scientific-technical domains. They developed a hybrid system that benefits from the different features of specialized documents to generate their summaries. Four models were integrated into this system: CORTEX (see section 3.7), ENERTEX (see section 3.9), the term extractor YATE (presented in section 6.3.1) and DISICOSUM, an algorithm based on a linguistic model developed at the IULA and which will be discussed below.

6.3.2.1. DISICOSUM

Da Cunha and Wanner's [CUN 05] hypothesis is that professionals in the medical domain use very concrete techniques for summarizing their texts. [CUN 05] studied a *corpus* composed of medical articles and their summaries to find out which information should be included in specialized summaries. Another starting point was the idea that different types of linguistic criteria should be used to create an adequate representation of the texts so that their advantages can be exploited. However, linguistics-based summarization systems generally use just one type of criteria (terms [LUH 58], textual position [BRA 95, LIN 97] or discursive structure [MAR 98, TEU 02], etc.), but do not combine them.

DISICOSUM [CUN 08] is constituted of rules based on textual, lexical, discursive, syntactic and communicative structures:

– textual rules: the summary must contain information from each section in the article. Sentences in certain positions are improved with an additional weight;

– lexical rules: the score of sentences containing words in the title, verbs conjugated in the first person plural, certain verbs (to analyze, observe, etc.) and nouns, numeric information in the sections "Patients and Methods" and "Results" are improved. Sentences containing references to tables, figures, statistics and computing, previous works and definitions are eliminated;

– rules combining discursive, syntactic and communicative structures: two theoretical frameworks were used: rhetorical structure theory (RST) [MAN 88] and meaning-text theory (MTT) [MEL 88, MEL 01]. Several examples of these rules are: [21]

- IF S is satellite CONDITION then maintain S:

[If these patients need a higher flow,]$_S$ [it is probable that it is not well tolerated.]$_N$

- IF S is satellite BACKGROUND then eliminate S:

[*People who do not want to eat, for fear of gaining weight, are anorexic.*]$_S$ [We studied the aspect of complications in anorexic patients.]$_N$

- IF S is satellite ELABORATION and S elaborates about the nucleus, then eliminate S:

[People who do not want to eat, for fear of gaining weight, are anorexic.]$_N$ [*One of the problems these patients have is low self-esteem.*]$_S$

[21] N = nucleus, S = satellite. Text in italics will be eliminated.

To establish these rules [CUN 09] used a Spanish segmenter,[22] the TREETAGGER system for POS tagging, as well as a manual discursive tagging of the *corpus*. In an annotation interface, the user chooses the relation between the different elements (nuclei and satellites) in the texts. The user detects the relations between sentences and the relations within the sentences (if they exist). The result is a representation of the document in the form of a relation tree. It is important to specify that the model's rules are applied to each section of the text separately. Figure 6.7 shows the architecture of the linguistic summarizer DISICOSUM.

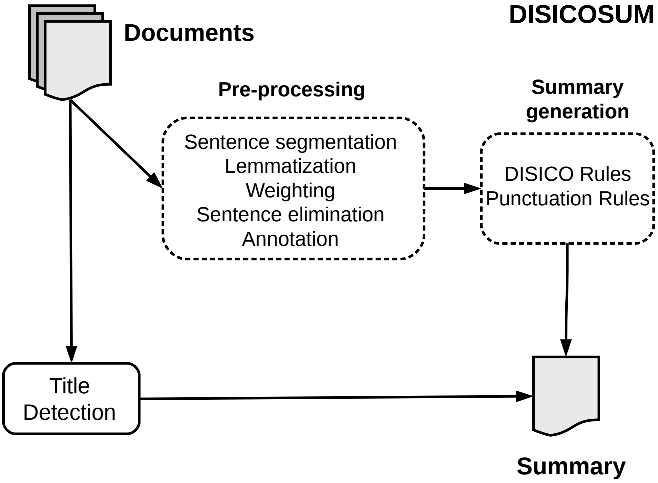


Figure 6.7. *Architecture of the linguistic summarization system* DISICOSUM

6.3.2.2. *A hybrid summarizer*

The experiments in this section focused on medical texts. Figure 6.8 shows the architecture of the hybrid system. The CORTEX, ENERTEX and DISICOSUM systems are applied to documents in the *corpus* in turn to weight the sentences. A decision algorithm processes the normalized outputs of the systems as follows:

1) agreement: extract a sentence selected by the three systems;

[22] Developed at the IULA.

2) majority: extract a sentence selected by the majority of the systems;

3) score: if a consensus cannot be reached, the algorithm extracts the sentence with the highest score;

4) score + the sentence order in the original text. If the scores are level, sentences are extracted according to their order in the text.

[CUN 09] arrived at the conclusion that the combination of CORTEX+YATE, ENERTEX and DISICOSUM obtained the best results. Therefore, their hybrid system is composed of these systems, as is shown in Figure 6.8.

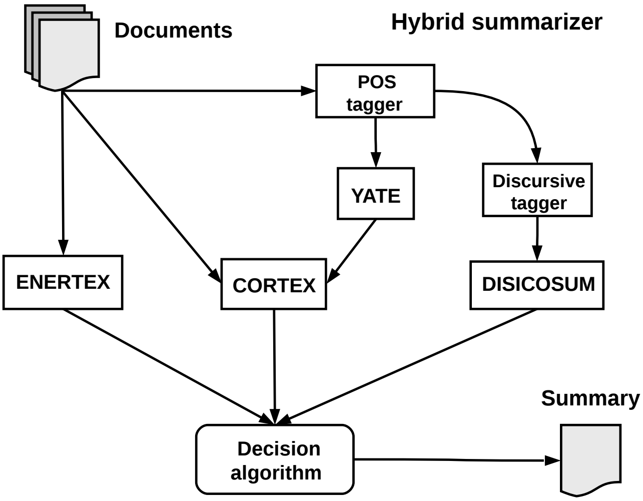


Figure 6.8. *General architecture of the hybrid system for biomedical summaries*

The *corpus* used for experiments contains medical articles written in Spanish from the journal *Medicina Clínica.*[23] The following systems were used to compare the performance of the hybrid system: CORTEX, ENERTEX, OTS, PERTINENCE SUMMARIZER, SWESUM,

[23] Each document comprises the sections “Introduction”, “Patients and methods”, “Results” and “Discussion”.

Word and two random baselines as well as the author abstracts which were used as references. Baseline 1 was constructed from the original article and baseline 2 was constructed from the reduced (by applying DISICOSUM's elimination rules) length article. The summaries were compared using ROUGE, with lemmatization and a Spanish stoplist.

Figure 6.9 shows the performance of the hybrid system in relation to the others. The summaries were truncated to 100 words, in accordance with the NIST protocol, to avoid favoring summaries containing sentences which were too long. The hybrid system performed extremely well and considerably outperformed all the other systems.

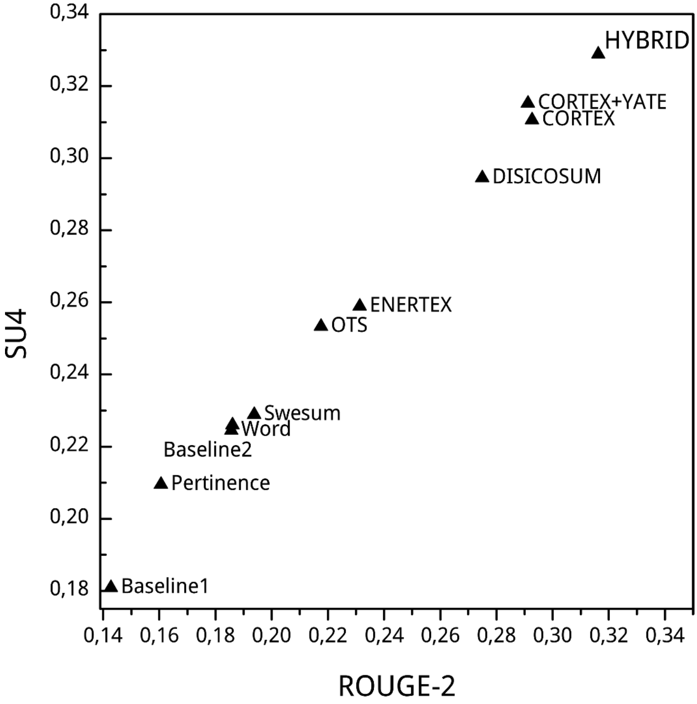


Figure 6.9. *Evaluation of the hybrid system on a biomedical corpus*

As an example, Table 6.2 shows the translation of the extract produced by the linguistic-statistical algorithm, the abstract written by the author of the article and the extract produced by one doctor. As is evident, the summaries are of different lengths. The summary produced by the algorithm is composed of two sentences, as it is the summary of the *Discussion* section of a medical article. However, the author summary contains just one sentence and the summary written by the doctor contains four sentences. It is worth noting that the two sentences selected by the algorithm (1a and 2a) were also retained in the doctor's extract (3c and 4c). Although the author of the article only puts forward one sentence in his summary (1b), it reflects a fusion of the ideas expressed in the two sentences of the summary produced by the algorithm (1a and 2a).

6.4. Summarizing court decisions

The automatic summarization of court decisions is a relatively unexplored and complex domain. Farzindar *et al.'s* work [FAR 04a, FAR 04b] concerns this type of summary. The authors compiled a corpus from 3,500 judgment documents from the Federal Court of Canada.[24] They manually analyzed 50 judgments in English and 15 in French, as well as their summaries which were written by professional legal summarizers.

A typical court decision document from this corpus has the following structure:

1) *decision data:* names, identities, titles, etc.;

2) *introduction:* who? did what? to whom?;

3) *context:* the facts in chronological order or by description;

4) *juridical analysis:* comments from the judge, the application of the law in accordance with the facts, etc.;

5) *conclusion:* the final court decision.

The modules of the LETSUM system (see Figure 6.10) realize the following tasks:

Thematic segmentation: segmentation is realized with a thematic segmenter. The segmenter is oriented to court decisions and takes a

[24] These documents are available at: http://www.canlii.org/en/ca/fct/.

certain number of linguistic markers for each section in the document into account;

Sentence	Extract generated by the algorithm
1a	The genotypic study of the generative region of the UGT-1 gene facilitates the identification of individuals with allelic variations associated with unconjugated hyperbilirubinemia and confirms the diagnosis of Gilbert's syndrome which is currently realized by excluding other pathological processes.
2a	These results enable the establishment of the convention of scrutinizing molecules for Gilbert's syndrome to be carried out via an additional test and the protocol for diagnosing chronic moderate intensity unconjugated hyperbilirubinemia, in the absence of hepatic illness in adults and newborns with prolonged neonatal icterus.
Sentence	Author abstract
1b	*The percentage of mutated alleles detected in the population we analyzed, similar to that found in other Caucasian populations, establishes the inclusion of the genotypic analysis of the UGT-1 gene in the protocol of diagnosing chronic moderate intensity unconjugated hyperbilirubinemia, in the absence of hemolysis and hepatic illness.*
Sentence	Extract produced by doctor 1
1c	*When we have a patient with indirect episodes of hyperbilirubinemia, we usually diagnose Gilbert's syndrome after having excluded the existence of a hepatopathy or hemolytic syndrome.*
2c	*In addition to being difficult for the patient, these tests are not very specific.*
3c	*The genotypic study of the generative region of the UGT-1 gene facilitates the identification of individuals with allelic variations associated with unconjugated hyperbilirubinemia and confirms the diagnosis of Gilbert's syndrome which is currently realized by excluding other pathological processes.*
4c	*These results enable the establishment of the convention of scrutinizing molecules for Gilbert's syndrome to be carried out via an additional test and the protocol for diagnosing chronic, moderate intensity unconjugated hyperbilirubinemia, in the absence of hepatic illness in adults and newborns with prolonged neonatal icterus.*

Table 6.2. *Examples (translations) of summaries generated by the hybrid algorithm, the author and a doctor (from [CUN 09])*

Filtering: this module identifies fragments in the document which could be eliminated without loss of relevant information, such as quotations (approximately 30% of the document). This module uses linguistic markers: verbs, concepts (nouns, adverbs and adjectives) and additional indications;

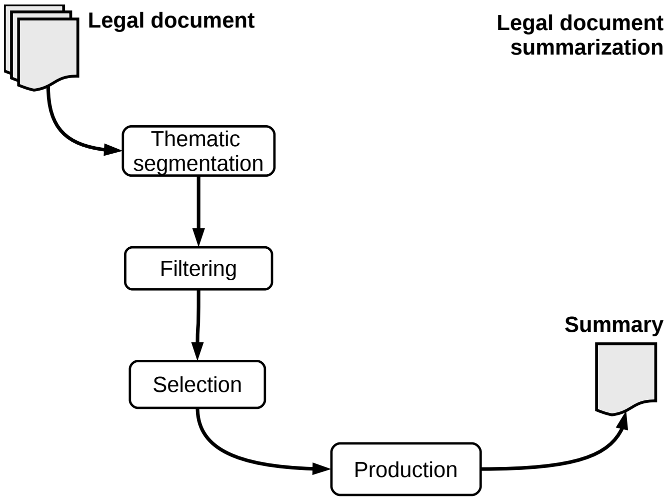


Figure 6.10. LETSUM, *summarization system for court decision documents*

Sentence selection: this module realizes sentence weighting by simple heuristics: position of the paragraphs in the document, position of the paragraphs in the thematic segment, position of the sentences in the paragraph, distribution of the words in document and corpus (tf.idf).

Generation of the summary: extracted sentences composing the summary are normalized and displayed in table format (summary). The length is approximately 10% of that of the source document.

The results on the test corpus show an F-measure of 0.90 in the thematic segmentation and an F-measure of 0.97 in the filtering stage. The results concerning summaries were more difficult to evaluate directly. On a subsequent and more complete system (based on the idea of LETSUM and using Unitex transductors[25]) [CHI 09], evaluators found that approximately 70% of sentences or paragraphs were correctly identified.

[25] http://www-igm.univ-mlv.fr/unitex/.

6.5. Opinion summarization

A task that has recently been studied is opinion summarization: systems which simultaneously detect an opinion (positive, negative or neutral) and produce summaries guided by this opinion. Producing this type of summary is a difficult task because of the considerable subjectivity of opinions. However, generic systems can be adapted to process this summarization task [BOS 09]. For instance, [LIU 09] developed an opinion summarization system for objects for sale. The solution he proposed was to divide the problem into three tasks:

– task 1: extracting the object's features from the comments;

– task 2: determining to what extent these features are positive, negative or neutral;

– task 3: clustering similar opinions and producing a summary.

This strategy can be adapted in a generic fashion to other opinion summarization problems. The pilot task of the TAC 2008 campaign, *opinion summarization (OS) task*, required very short summaries to be produced which reflect the opinions of a given entity (names of people, names of organizations, products, etc.) on a corpus of blogs.

In the following section, the CBSEAS system, which participated in this OS task, will be presented.

6.5.1. *CBSEAS at TAC 2008 opinion task*

The multidocument summarization system CBSEAS [BOS 09] was presented in section 4.6.5. Its developers modified the system to enable it to process the polarity of opinions. More specifically, each sentence in the corpus was assigned a positive or a negative polarity. Two support vector machine (SVM) classifiers were trained to classify the queries (SVM_2) and the sentences in the blogs to be summarized (SVM_1). Detecting the polarity of queries enables, in principle, summaries to be better oriented toward user expectations. Figure 6.11 shows the modifications made to the original system.

To test this system, the TAC 2008 OS corpus (from Blogs06 Text Collection) was used: 25 clusters and targets (i.e. target entity and questions). The objective was to produce a short (approximately 7,000 characters), well-organized and readable summary, which was evaluated using the semi-automatic PYRAMID metric[NEN 07].

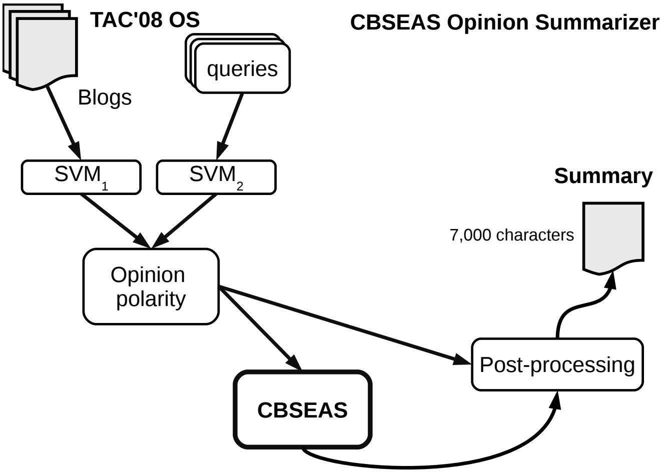


Figure 6.11. *CBSEAS system for opinion summarization*

The authors observed that it is easier to classify queries because they have regular patterns which can be deduced from the previous NIST campaigns. Conversely, the considerable variability of the blogs makes it difficult to classify them. To overcome this problem, they opted for the following strategy: to give each sentence the polarity of the entire blog post. This decision was based on three criteria:

– blog posts rarely offer a nuanced opinion and often put forward a positive or a negative opinion;

– several sentences are relevant though they do not express an opinion;

– several sentences require context to enable them to be classified.

Table 6.3 shows the results obtained by this system. As it can be seen, in general it is well ranked, particularly in structure, fluency and grammaticality.

Evaluation	CBSEAS	Rank
Pyramid	0.169	5/20
Grammaticality	5.950	3/20
Non-redundance	6.636	4/20
Structure	3.500	2/20
Fluency	4.455	2/20
Responsiveness	2.636	8/20

Table 6.3. *Evaluation of the CBSEAS system for the TAC 2008 OS task*

6.6. Web summarization

The Web can be considered a huge *corpus:* it is heterogeneous, noisy and composed of documents of different genres, sources and size. Of course, the web is not a *corpus* in the strict sense of the term, rather it is a huge warehouse full of data, which is often unstructured. Given the considerable redundancy on the web, automatic text summarization techniques could be useful for filtering information. In this section, three types of automatic summarization for online documents will be studied:

– web page summarization [BER 00];

– microblog summarization (*tweets*) [LIU 12, CHA 11];

– email summarization [CAR 07].

6.6.1. *Web page summarization*

The rapid growth of the internet has generated an enormous ocean of web pages. Sites such as Yahoo! or the Open Directory Project (DMOZ),[26] organize web pages in a hierarchical structure and provide a short summary of each page to facilitate access. Google and other search engines produce, for each query, a list of uniform resource locators (URLs)[27] accompanied with a *snippet* of each page, which gives a brief overview of the content of each page.

[26] http://www.dmoz.org/.

[27] "A uniform resource locator, abbreviated as URL (also known as web address, particularly when used with Hypertext Transfer Protocol (HTTP)), is a specific character string that constitutes a reference to a resource" (source: Wikipedia: http://en.wikipedia.org/wiki/Uniform_resource_locator).

However, manually maintaining the pairs {web pages/summary} in an organized structure cannot keep up with the exponential growth of the web. A *snippet* is not enough: it is just a small portion of a web page which is generally limited to several characters at the beginning of the page. Automatic text summarization techniques can be applied to this type of document. Thus, we can speak about extraction- or abstraction-based web page summarization as well as generic or query-guided (by a user) summarization.

Despite the effort made to solve this task, the results are still not satisfactory. Indeed, the textual content of web pages can be rather empty: the page is often "polluted" with non-textual content (images, videos, sounds, executable codes, etc.) making analysis difficult. Moreover, the content of the page can include a large number of topics. As a result, classic automatic summarization methods are often incapable of focusing on important topics or calculating the relevance of the text fragments.

6.6.2. *OCELOT and the statistical gist*

Research into web page summarization uses for the most part data from the website DMOZ. In this directory, couples – web page/summary – are organized so that the pages with the most general topics are at the top of the hierarchy. Summaries are written by the community and are a precious resource which serve to evaluate the performance of the systems. Berger & Mittal's article [BER 00] presented the OCELOT system for web page summarization. Though a classic article on the subject, it is very educational and the results it presents are very interesting. Figure 6.12 shows the architecture of the summarization system OCELOT.

The authors used a machine translation (MT) approach to produce summaries: the noisy channel model.[28] It is an intuitive model which

[28] "The noisy channel model is a framework used in spell checkers, question answering, speech recognition, and MT. In this model, the goal is to find the intended word given a word where the letters have been scrambled in some manner" (source: Wikipedia: http://en.wikipedia.org/wiki/Noisy_channel_model).

has also been used in other NLP tasks. The authors used this model to align the summaries (gists) with the content of web pages in DMOZ.

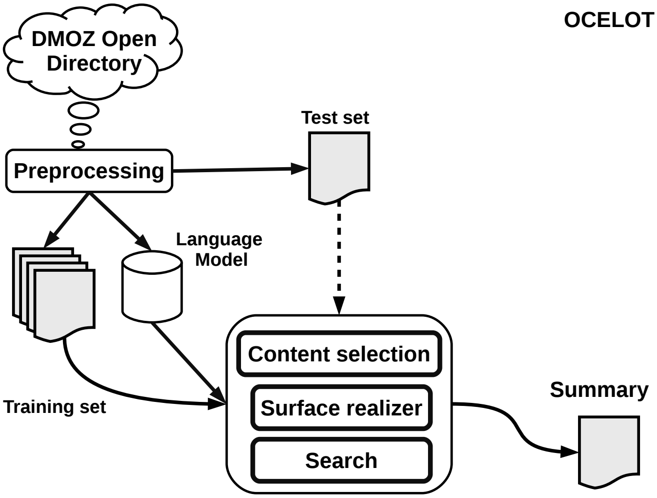


Figure 6.12. OCELOT *system*

The idea is as follows: to postulate a probability model $p(g|d)$ which assigns a probability based on the fact that the string of words $g = \{g_1, g_2, ..., g_n\}$ is the best gist of the document $d = \{d_1, d_2, ..., d_m\}$. Given a document (source) d, the optimal gist $g*$ for this document is:

$$g* = \arg\max_g p(g|d) \qquad [6.3]$$

$$= \arg\max_g \frac{p(d|g)p(g)}{p(d)} \qquad [6.4]$$

and finally,

$$g* = \arg\max_g p(d|g)p(g) \qquad [6.5]$$

where $p(d|g)$ is a model which measures the proximity of g and d, and $p(g)$ measures readability (the *a priori* coherence of the gist) through a language model.

Then, to generate the gist of a web page, three models are applied:

1) content selection: determining which words will be used in the summary;

2) word ordering: arranging these words so that they are pleasing to read;

3) search: finding a doubly optimal sequence of words in terms of content and readability.

The corpus of web pages was divided into two subsets: a training set of 99% and a test set of 1%. As stressed by the authors, evaluating algorithms for MT traditionally requires a much higher number of test examples. However, 1% of the open directory dataset is estimated to be several thousands of web pages, which is broadly sufficient for the purpose of the article. The training set served to build a language model and a characteristic learning model.

In OCELOT, the stage of preprocessing web pages was taken much further than in classic text summarization. The text is filtered and normalized several times, as indicated in [BER 00]:

– remove punctuation, convert all text to lowercase; replace numbers by the label NUM;

– remove the most common stopwords (using a stoplist);

– remove all hyperlinks, images, Hyper Text Markup Language (HTML) markup and meta-information from the web pages;

– remove pages containing adult-oriented contents;

– remove the pages with frames;

– eliminate web pages and gists which are too short (< 400 or < 60 characters, respectively);

– remove duplicate web pages;

– remove pages which were just “404 Not Found”.

The summaries generated are one sentence in length, which is also useful for title generation. Summaries can contain words which are not present in the source web page, which is interesting. Moreover, by taking advantage of ML techniques, OCELOT has the capacity to take an input web page in one language and to produce the gist in another language (tests were reported for the language pair French into English).

The evaluation of the OCELOT system was realized via word overlap (WO), the perplexity of the language model and the direct reading of summaries. WO was defined as:

$$\text{WO} = \frac{\sum_w w \in \text{gist} \cap w \in \text{summary}}{|\text{summary}|} \qquad [6.6]$$

WO measures the fraction of words in the summary which appear “appropriate” for describing a gist. Table 6.4 shows the results of this evaluation.

Summary length	Word overlap (WO)
4	0.4166
5	0.3489
6	0.3029
7	0.2745
8	0.2544
9	0.2353

Table 6.4. *Evaluation of bag-of-words in the* OCELOT *system*

An example of a gist from the open directory dataset and the corresponding summary produced by OCELOT:

DMOZ gist: `advocate the rights of independent music artists and`
`raise public awareness of artists distributing their music`
`directly to the public via the internet`

OCELOT: `music business and industry artists raise awareness rock`
`and jazz`

6.6.3. *Multitweet summarization*

Twitter[29] is a social network with a microblog type service, where a large number of users[30] continually exchange information in the form of tweets[31]. The network has exponentially grown since 2006 and the huge volume of information (users post 340 million tweets per day) is available in real time.

Here are several examples of tweets concerning the query “giants rats in England”:

1) `@PeterPeinl 2h. Mutant giant kangaroo invade Germany !! lol!:))`

2) `@sixtine 17 apr Giant Rats in england, they are big like a kat W-H-A-T? How am I gonna do, I coming in months!! #beurk #rats #invasion`

3) `@LennyLovet 15 apr. @BBCRadio It’s a member of The RockN Roll Rats!`

4) `@Darcy 15 apr. RT @Kirstie 14 apr. Very Huge rats in England!!! Danger!!!: http://pic.twitter.com/JWfgaXtts`

5) `@BBCRadio 16 apr. Giant rat in Gravesend pic.twitter.com/JWfgdXtpK, Why is England going back to how it was in 1600s? We are not prepared to deal with spreading giant rats`

6) `@Kirstie 16 apr. @BBCRadio @Independent yes no problem:)`

7) `@BBCRadio 16 apr. Hi @KristieRol Hi @GuardianUS, is it okay if we use your pictures on our Facebook page? We’re discussing Hampshire rats today. Thank you x`

8) `@Independent 16 avr. @Kirstie Hi, could`

[29] http://www.twitter.com.

[30] Twitter had over 255 million monthly active users (MAUs) in March 31, 2014. Source: https://investor.twitterinc.com/releasedetail.cfm?ReleaseID=843245.

[31] A type of short message service (SMS) which is less then 140 characters.

`we have permission use your picture for @Independent? Thanks`

9) `@GuardianUS 14 apr. A 2 ft-long rat was caught in England, but human civilization shouldn't panic (yet) http://pic.twitter.com/JWfgdXtpK`

10) `@Kirstie 14 apr. Very Huge rats in England!!! Danger!!!: http://pic.twitter.com/JWfgaXtts`

The task of multitweet summarization aims to find one or several tweets which represent a given set of tweets. In the previous example of a set of tweets, tweets 1 and 3 do not really have anything to do with the topic. Tweet 4 is a retweet of 5; tweets 6, 7 and 8 are private exchanges. Therefore, multitweet summary could be the following:

– `@BBCRadio 16 apr. Giant rat in Gravesend pic.twitter.com/JWfgadXtpK, Why is England going back to how it was in 1600s? We are not prepared to deal with spreading giant rats`

– `@GuardianUS 14 apr. A 2 ft-long rat was caught in England, but human civilization shouldn't panic (yet) http://pic.twitter.com/JWfgadXtpK`

– `@Kirstie 14 apr. Very Huge rats in England!!! Danger!!!: http://pic.twitter.com/JWfgaXtts`

This type of summary enables users to quickly understand the essential meaning of a large number of tweets. It can also be used as a component of preprocessing for information extraction tasks on tweets. The challenge of this task is to calculate the tweet's relevance score from the very limited information it contains. The use of Hidden Markov Models enables the temporal events of sets of tweets to be modeled [CHA 11]. In the following section, an approach to multitweet summarization which uses a graph-based representation will be presented.

6.6.3.1. *Graph-based multitweet summarization*

The graph-based multitweet summarization system [LIU 12] integrates the functionalities of social networks, enabling the lack of information contained in tweets to be confronted.

A first system was defined as follows:

Preprocessing: a preprocessing operation was carried out and each tweet was represented as a bag-of-words. The metadata of tweets (hashtags, usernames and links) were normalized: usernames → ACCOUNT, links → LINK. Hashtags were processed as keywords and misspelt words were corrected using a dictionary.

Tweet weighting: the graph-based automatic text summarization system LEXRANK [ERK 04] (see section 3.5.4.1) was chosen as a base system, thanks to its performance during the DUC evaluations.

Tweet selection: the selection function is an implementation of the maximal marginal relevance (MMR) algorithm [CAR 98] (see section 4.5.2), which selects the most M tweets as the summary. The choice of a tweet depends on two factors: its relevance score and its resemblance to tweets which have already been selected.

Therefore, this first system constitutes a generic multitweet summarizer. To improve it, three important components were added to the system:

Social network features: retweets and the number of followers are used to obtain a more reliable relevance value.

Readability: the introduction of a readability function to make the summary more pleasant to read.

Diversity: user diversity – including a tweet only if the number of tweets selected from the same Twitter account does not exceed a fixed limit.

The readability module is based on the Flesch formula[32] (equation [6.7]), extended by the number of out of vocabulary (OOV) words and

[32] http://en.wikipedia.org/wiki/Flesch-Kincaid_readability_tests.

the number of abnormal symbols.

$$206.835 - (1.015 \times ASL) - (84.6 \times ASW) - (10.5 \times AOW) - (9.8 \times AAS) \quad [6.7]$$

where ASL is the average sentence length (number of words divided by number of tweets), ASW is the average word length in syllables (number of syllables divided by number of words); AOW is the average number of OOV words (the number of OOV words divided by the total number of words) and AAS the average number of abnormal symbols (the number of abnormal symbols divided by the total number of words).[33]

Figure 6.13 shows an overview of Liu *et al.'s* system [LIU 12].

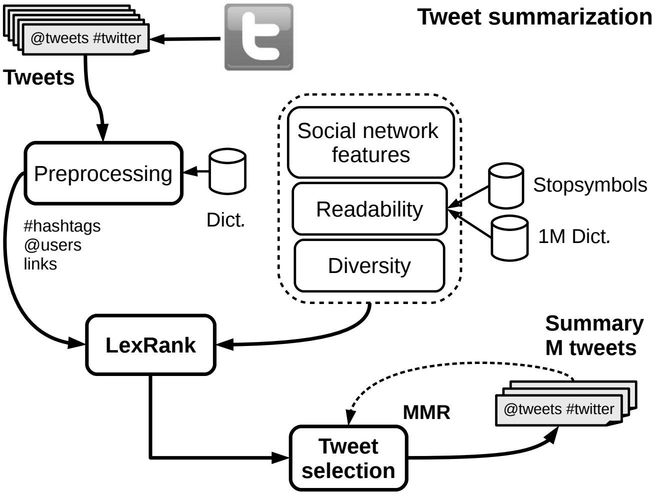


Figure 6.13. *Graph-based tweet summarization system*

To compare its results, the authors defined the baseline as the generic summarization system. The system was evaluated on a set of manually

[33] The authors compiled a dictionary of 1M OOV words and 125 abnormal symbols.

annotated data (gold standard summary). It outperformed the baseline in ROUGE-1 and -2 scores (see Table 6.5).

System	ROUGE-1	ROUGE-2
Baseline	0.3591	0.2871
[LIU 12]'s system	*0.4562*	*0.3692*

Table 6.5. ROUGE *evaluations of multitweet summarization systems*

6.6.4. *Email summarization*

The recent task of email summarization faces several difficulties. First, the content of emails is far from being structured into sentences. Users communicate through badly written messages containing sentences which are too short or too long. In emails people write much too freely and, with some exceptions, messages are very short. Another difficulty is linked to email threads. In fact, dialogues via email contain the whole or fragmented content of messages in replay which intermingle in a very complex fashion.

Work into automatic email summarization can be divided into two categories:

1) summarization of individual emails;

2) summarization of email threads.

An overview of the task of email summarization will be given as an in-depth study falls beyond the scope of this work.

The Enron Corpus is very commonly used in the two tasks. It is composed of approximately 0.5 M of electronic messages. It contains data from some 150 users, principally from the upper echelons of the company Enron. [34] FoundationDB [35] produced a Python tutorial which

[34] The Enron Corpus is available at: http://www.cs.cmu.edu/./enron/, [KLI 04] Klimt B., Yang Y., "The Enron Corpus: a new dataset for email classification research", *Machine Learning: ECML 2004*, Springer, Berlin/Heidelberg, vol. 3201, 2004.

[35] https://foundationdb.com.

describes modeling and accessing the Enron corpus. It is available at https://foundationdb.com/documentation/enron.html#enron-appendix.

The GIST-IT algorithm was one of the first works into individual email summarization [MUR 01]. They use a combination of linguistic approaches to extract the nominal syntagma and then machine learning approaches to give a degree of relevance to the extracted nominal syntagma. A qualitative evaluation showed that in the GIST-IT system, nominal sentences are better candidates than n-grams for the representation of email content.

With regards to email threads, the Clue Word Summarization (CWS) method uses email quotations in graph form [CAR 07]. The authors calculate a graph where each email is mentioned by other emails at the sentence and segment level. *Clue Words* are semantically similar words which appear simultaneously in a father node and its sons. Sentences are weighted according to the number of clue words they contain.

CWS' results in precision are higher than those obtained by MEAD and RIPPER [RAM 04], as shown in Table 6.6.

System	Sentence-level	Conversation-level
MEAD	0.200	0.225
RIPPER	0.330	0.400
CWS	*0.400*	*0.350*

Table 6.6. *Performance of systems for the email summarization task*

6.7. Conclusion

In this chapter, the difficulties linked to generating summaries of documents from specialized domains or different sources were presented. Terms are frequently used in specialized domains; however, the concept of *term* itself is often ambiguous and depends on the specialization. Three systems were presented: YACHS2, dedicated to obtaining the summaries of organic chemistry texts; SUMMTERM, a semantic summarizer; and a linguistic-statistical hybrid algorithm, both of the latter oriented at biomedical documents. YACHS2 wants to remain independent of domain-specific resources and, therefore, favors statistical approaches applied to chemistry texts written in English. This system uses a specific preprocessing operation to protect the names of chemical entities (detected by learning and rules) before their normalization and sentence weighting. The latter is carried out not only with classic measures but also with similarity at the character level, which enables commonly used symbols in the nomenclature of chemistry to be detected at a fine level. The SUMMTERM system combines terminological (the term extractor YATE) and semantic (the *EWN* ontology) resources to summarize biomedical documents in Spanish. It obtained good results in this task: both producing and evaluating specialized summaries is difficult. This method could be applied to other domains, supposing that EWN has the necessary terms. Terms can be weighted by classic tf.idf or other more complex mechanisms, such as a term extractor. This is the case for the YATE extractor, combined with CORTEX in the linguistic-statistical model. The hybrid system obtains very good results, even better than those obtained by each method separately. Fusing statistical methods with a linguistic algorithm showed that this approach is very interesting as it produces summaries which are closer to those expected by a specialist user.

Summaries from online documents (web pages, tweets and email) pose considerable challenges to systems. Texts that are empty, not written in sentences, too short or too long, unstructured and polluted with non-textual sources, etc. require a specific preprocessing operation before being analyzed and summarized. However, the automatic summarization methods seen in the previous chapters can easily be adapted to these new tasks. Results are still mitigated, leaving room for improvement.

7

Text Abstracting

Research into automatic text summarization has principally focused on extraction-based systems. However, the general trend is increasingly leaning toward the generation of abstraction-based systems. Indeed, the Document Understanding Conference (DUC)/Text Analysis Conference (TAC) campaigns related to guided summarization want to encourage research into these types of summarization systems. In this chapter, several systems for abstract generation will be presented. For instance, SumUM and Summarist use natural language generation (NLG) techniques; RIPTIDES combines information extraction (IE), extraction-based summarization and NLG to produce multidocument summaries. The state of the art shows that multisentence compression (MSC) and fusion enable us to get a step closer to abstract generation. Therefore, two approaches to elementary paraphrasing and sentence fusion and compression will be presented.

7.1. Abstraction-based automatic summarization

As has been seen in the previous chapters, extraction-based summarization has been very successful in determining which sentences in the input documents contain important information. However, it has also been seen that these methods are very far from producing optimal summaries, both in terms of content and linguistic quality. Problems of cohesion, coherence and resolving anaphoras prevent extraction-based systems from producing quality outputs. The HexTAC experiment (see sections 2.6.2 and [GEN 09a]) and other studies on professional summarizers [CRE 93, CRE 96] have shown that, unlike the majority of extraction-based systems, people seek to produce summaries by abstraction. In abstracts, paraphrasing, rewriting complex sentences and introducing new linguistic information seek to produce a concise version of the content. Although people re-use parts of the source document, they do not re-use all of it: they use segments or parts of sentences instead of using the whole sentence to produce the summary. A great deal of research has been carried out into extraction-based algorithms, but very few works exist in the context of abstraction-based summarization.

Abstraction-based approaches seek to produce summaries whose quality is comparable to summaries produced by people. To arrive at this point, a certain number of algorithms have been explored. For instance, Jing's [JIN 99] work showed that the two most important operations for people during the generation of summaries are sentence compression and combination. Methods using sentence compression [KNI 02, YOU 07, MOL 13c] and information fusion [FIL 10, BOU 13] have, therefore, been explored. Other algorithms, such as title generation methods (ultra-summarization) [BAN 00] as well as methods using NLG modules [HOV 98, LIN 98, HOV 99, WHI 01, SAG 02b], have also been used in the context of abstraction-based summarization.

7.2. Systems using natural language generation

Generative summarization systems often use extraction methods and specialized modules from NLG. It must be stressed that NLG is an extremely complex and relatively unexplored task; it falls outside the scope of this work [1]. However, a brief overview of the modules that form the basis of NLG will be given below. Two components are necessary for text generation:

The microplanner: this converts a specification at the discourse level of a sequence of subjects into a list of specifications at the level of (several) clauses. This involves choosing between several aspects, such as sentence length, the internal organization of the sentence (prepositions, active or passive voice etc.), identification of the main theme, choice of the main verb and other important words, among

[1] If interested, please consult [REI 00] and [MCK 85].

others. The microplanner must also ensure that the information selected by the topic identification and interpretation modules be formulated in a compact manner and be as brief as possible, while remaining grammatical.

The sentence generator: the sentence generator (or text realizer) converts a relatively detailed specification of one or several clauses into a grammatical sentence.

In the SUMMARIST system [HOV 98, LIN 98, HOV 99] (presented in section 5.4), the process of text generation uses topic identification and interpretation to produce an abstract. The microplanner and sentence generator modules were not developed, rather they were borrowed from other research. The authors recognize that these tasks remain largely misunderstood and that the results are no better than those produced by the extraction-based module in SUMMARIST. Figure 7.1 shows SUMMARIST's NLG modules.

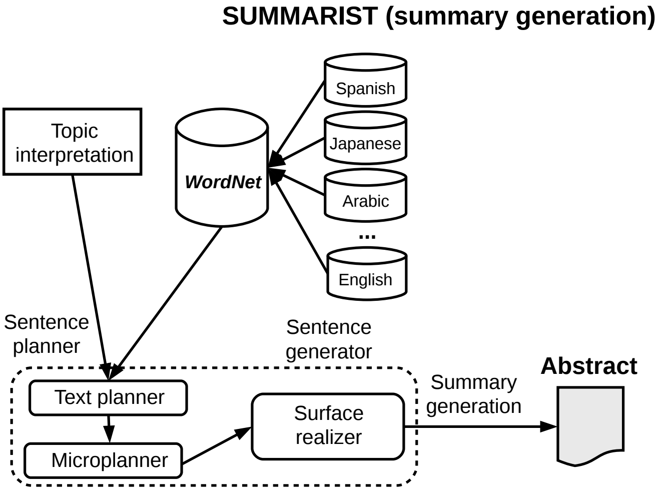


Figure 7.1. *Abstract generation in the* SUMMARIST *system*

On the other hand, SUMMUM is a multidocument system, presented in section 4.5.7, which is capable of producing indicative and informative summaries. Indicative–informative summaries are generated from certain topics. As pointed out by [LAP 13, SAG 02b]: "SUMMUM motivates the topics, describes entities, and defines concepts. This is accomplished through a process of shallow syntactic and semantic analysis, concept identification, and text regeneration".

7.3. An abstract generator using information extraction

In the RIPTIDES system [WHI 01], users select a set of documents from which the system extracts information in order to activate one of several scenario templates (extraction domains). Users provide filters and preferences for the scenario templates. They must specify the information they want to appear in the summary.

The architecture of the RIPTIDES system is shown in Figure 7.2.

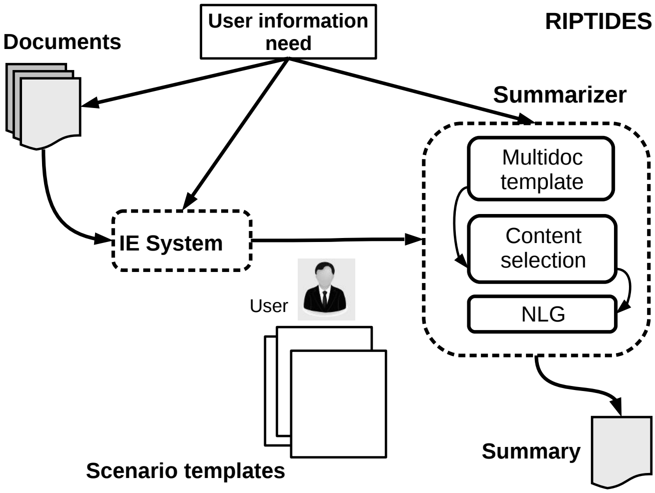


Figure 7.2. RIPTIDES *summarization system*

With regard to evaluation, the corpus used was Topic 89 from the TDT2 collection: 25 documents about the 1998 earthquake in Afghanistan. Each document was manually annotated with a natural disaster scenario, including the expected output by the IE system.

Two human summarizers, RIPTIDES and a baseline produced summaries according to the instructions. The baseline system was an automatic extraction-based multidocument summarizer, which took the position of the sentence in the document and word clusters into account. Each summarizer produced different types of summaries:

1) *a 100-word summary over 2 days;*

2) *a 400-word summary [...] emphasizing specific and factual information;*

3) *a 200-word summary [...] focusing on the damage caused by the earthquake, and excluding information about relief efforts;*

4) *a 200-word summary [...] focusing on the relief efforts, and highlighting the Red Cross's role in these efforts.*

Three judges evaluated the performance (on a scale of A – the best – to F – the worst –) of the summarizers (systems and humans). The baseline obtained a score of D, RIPTIDES C and the two people A and B+. Overall performance as well as performance in terms of content, organization and readability (averaged over a numerical scale) are displayed in Table 7.1.

System	**Baseline**	**RIPTIDES**
Overall	0.96 ± 0.37	2.10 ± 0.59
Content	1.44 ± 1.00	2.20 ± 0.65
Organization	0.64 ± 0.46	2.08 ± 0.77
Readability	0.75 ± 0.60	2.05 ± 0.65

Table 7.1. *Manual evaluation of summaries generated by RIPTIDES [WHI 01]*

7.4. Guided summarization and a fully abstractive approach

One of the main objectives of the guided summarization task at the TACs was to encourage the development of abstraction-based approaches. Genest and Lapalme's research [GEN 12] was inspired by abstraction-based systems using IE to produce abstracts. Indeed, the idea of using IE to generate abstracts has been exploited by systems such as FRUMP [DEJ 82] (see section 2.4.1) or the RIPTIDES system [WHI 01] in the context of natural disasters. However, in [GEN 12], the authors used modern IE techniques, such as syntactic analysis, the recognition of named entities, NLG etc.

The algorithm realizes a relatively advanced preprocessing operation on the input documents:

– preprocessing with GATE [CUN 11].Filtering and normalizing the input by using regular expressions, sentence segmentation, tokenization and lemmatization;

– analysis of dependency and syntactic analysis using the Stanford parser [DE 08];

– named entity recognition (NER) using the Stanford NER system (see Appendix A.1.4);

– date resolution engine: disambiguation of days of the week and relative temporal terms.

Figure 7.3 shows the architecture of the abstract generation system by IE.

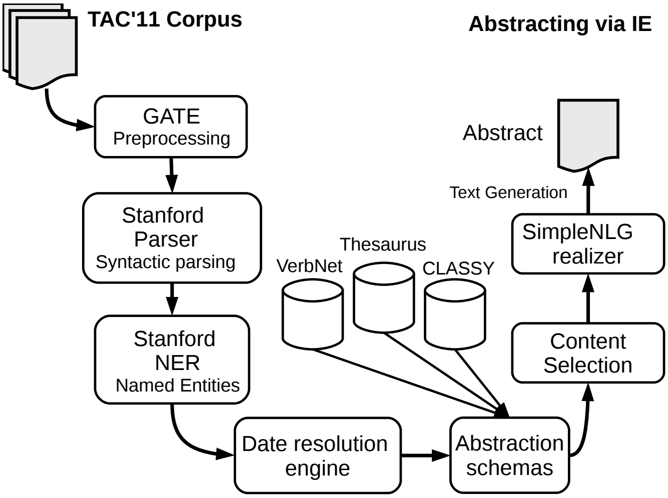


Figure 7.3. *Abstract generation system by IE [GEN 12]*

Once preprocessing is complete, the system uses abstraction schemas. An abstraction schema is a manually created structure composed of IE rules, heuristics for content selection and generation patterns. The abstraction schemas were designed in the context of a theme or a subtheme. An abstraction schema aims to respond to one or several aspects of its theme. The schemas' extraction rules depend on verbs and nouns (that have a similar meaning) and the syntactic position of the roles (for instance, the victim or the assassin in the theme "killing"). Indeed, a list of keywords is used to identify the words that imply "attack".

Three resources were necessary to elaborate extraction rules:

1) a thesaurus for finding semantically linked nouns and verbs;

2) VerbNet[2] to capture the semantic roles of the verbs and their syntactic dependents;

3) a manual list of the stems of the relevant words.

The content selection module receives and selects the candidates found by the IE rules for each aspect. The basic idea consists of choosing the most quoted candidate for an aspect. It is a complex module, so several heuristics are used to avoid redundancy and the least informative responses. Finally, the text generation module works as follows. The production models for each schema are simple. They are implemented with the SimpleNLG realizer[3], which takes as input a sentence structure and the lemmatized words and makes as output a sentence with resolved agreements. Text generation requires a text planner. The text planner in this system is composed of the ranking of the schemas. For instance, a summary concerning the theme "attack" starts with the schema "event". The redundancy of the schema is also managed via this ranking.

The authors carried out a small-scale evaluation concerning the quality of the generated abstracts. They used the PYRAMID metric on the TAC'11 corpus. They calculated a raw PYRAMID score of 0.54. This is remarkable, because only three of the 43 participating

[2] VerbNet is a resource developed by: [KIP 08], and is available for download from http://verbs.colorado.edu/mpalmer/projects/verbnet.html.

[3] http://www-etud.iro.umontreal.ca/vaudrypl/snlgbil/snlgEnFr_francais.html.

automatic summarizers beat this score in 2011 (the average was 0.31). It must also be stressed that the abstracts produced were just 21 words in length, compared to the 100 words of the summaries generated by the participants. This still leaves room for improvement in terms of linguistic quality and content.

7.5. Abstraction-based summarization via conceptual graphs

Miranda-Jimenez *et al.*'s research [MIR 13] developed a single-document summarization model by abstraction based on the conceptual representation of the text. This model uses a complete semantic representation of the text structures by conceptual graphs (CGs). As such, the task of generating a summary is reduced to summarizing all the corresponding CGs. CGs [4], based on logic, enable knowledge to be represented in a structured form. The texts are, therefore, represented in a detailed semantic form. Miranda-Jimenez *et al.* [MIR 13] introduced weighted CGs, where the arcs are weighted to produce a "semantic flow". The summarization algorithm uses a modified version of the Hyperlink-Induced Topic Search (HITS) method presented in [MIH 05b] for node weighting.

The architecture of the system based on CGs is shown in Figure 7.4.

Though an interesting approach, it must be said that the text generation module was not implemented in Miranda-Jimenez *et al.*'s work [MIR 13]. However, some work has been put forward for text generation from CGs. The algorithm has been tested on the DUC 2003 collection. Indeed, it appears that this method is well-adapted for processing short texts. A baseline was defined as follows: it selects the first concepts in the document until the compression rate is reached. The average results (Table 7.2) show that the algorithm for CGs is better than the baseline in terms of precision, recall and F-measure.

[4] "Conceptual graphs (CGs) are a formalism for knowledge representation. [...] John F. Sowa used them to represent the conceptual schemas used in database systems. The first book on CGs, [JOH 84], applied them to a wide range of topics in artificial intelligence, computer science, and cognitive science" (source: Wikipedia: http://en.wikipedia.org/wiki/Conceptual_graph).

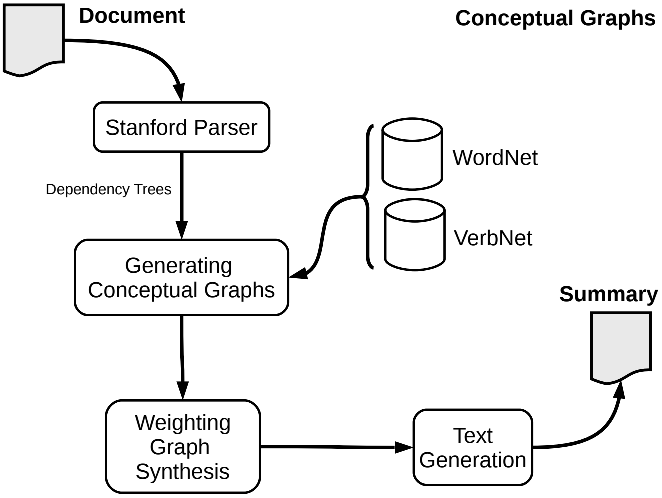


Figure 7.4. *Summarization system by conceptual graphs*

System	Baseline	Conceptual graphs
Precision	0.51	**0.62**
Recall	0.51	**0.70**
F-measure	0.51	**0.65**

Table 7.2. *Results from the algorithm for conceptual graphs (CGs)*

7.6. Multisentence fusion

An interesting strategy for reducing redundancy is the fusion of semantically similar sentences. This task was introduced in 2005 by Barzilay and McKeown [BAR 05] in the context of multidocument summarization: a set of sentences from different documents, which say more or less the same thing, is fused and compressed according to their resemblance. Indeed, the more documents there are about the same topic, the more important the redundancy of the generated extract. Thus, the objective is to reduce redundancy, while maintaining the essential information contained in a group of sentences. Multi-sentence fusion (MSF) (or MSC) is a task that consists of generating the simple version of a sentence from the fusion of a group of related sentences.

A definition of MSF is

DEFINITION 7.1.– *Multi-sentence fusion is a task that consists of generating the simple version of a sentence s_f from a set $\mathcal{C}$ of k related sentences $\{s_1, s_2, \ldots, s_k\}$. Sentence s_f:*

– retains the important information from set $\mathcal{C}$;

– is not necessarily part of set $\mathcal{C}$;

– is grammatically correct.

Sentence fusion can also be seen as an elementary form of paraphrasing. It can, therefore, be used to fuse and produce new sentences in the summary, which are not present in the source corpus.

Two approaches to sentence fusion will now be presented: that of Filippova [FIL 10] and of Boudin and Morin [BOU 13].

7.6.1. *Multisentence fusion via graphs*

[FIL 10] presents an unsupervised method for sentence fusion. The idea is to represent candidate sentences for fusion in an incremental word graph. This graph is then browsed to detect the terms (full words) with the greatest number of outgoing edges to create the fused path. [FIL 10] uses optimization methods to find the best path from the input to the output by browsing the heaviest outgoing nodes. Below, an example will be presented of the working of the algorithm for MSF applied to the following *corpus*:

1) The raw fish sushi was invented in Japan in the 16th Century.

2) The fish sushi was invented by the people of Japan.

3) In Japan, a new food based on fish, was invented.

4) Sushi, a Japanese invention in 1500.

5) The creation of sushi by Japanese in the 1500s.

Figure 7.5 shows the initial graph, constructed with the first sentence (where an initial point ● and a final point ○ were added to the graph):

● ⇝ the ⇝ raw ⇝ fish ⇝ sushi ⇝ was ⇝ invented ⇝
in ⇝ japan ⇝ in_the ⇝ sixteenth ⇝ century ⇝ ○

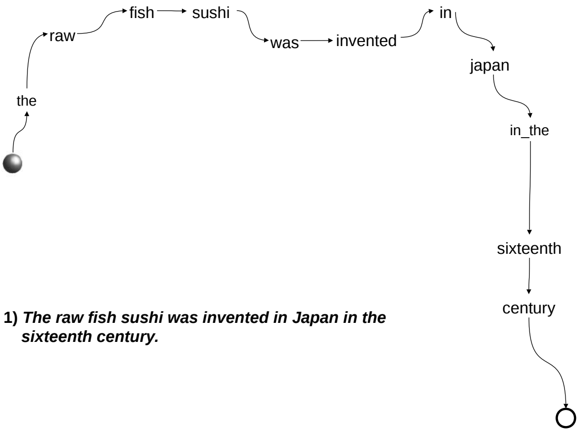

the
raw
fish
sushi
was
invented
in
japan
in_the
sixteenth
century
1) The raw fish sushi was invented in Japan in the sixteenth century.

Figure 7.5. *Multisentence fusion: initial step with the first sentence*

Figure 7.6 shows a modified graph on which the second sentence has been superimposed:

● $\rightsquigarrow$ `the` $\rightsquigarrow$ `fish` $\rightsquigarrow$ `sushi` $\rightsquigarrow$ `was` $\rightsquigarrow$ `invented` $\rightsquigarrow$ `by_the` $\rightsquigarrow$ `people` $\rightsquigarrow$ `of` $\rightsquigarrow$ `japan` $\rightsquigarrow$ ○

In this phase, it can be stated that the terms: $\mathcal{L} = \{$`fish, invented, japan`$\}$ each have two outgoing or ingoing edges. The process is iterated by adding the remaining sentences in turn. This gives the graph in Figure 7.7. In this graph, the set of the terms (vertices) of $\mathcal{L}$ has been reinforced with a greater number of outgoing edges. From then on, a path from input ● $\rightsquigarrow$ $\{$terms $\in \mathcal{L}\}$ $\rightsquigarrow$ ○ to output, in addition to some linking words, can be constructed. The fused sentence is finally:

`sushi was invented in japan.`

which is not part of the original corpus.

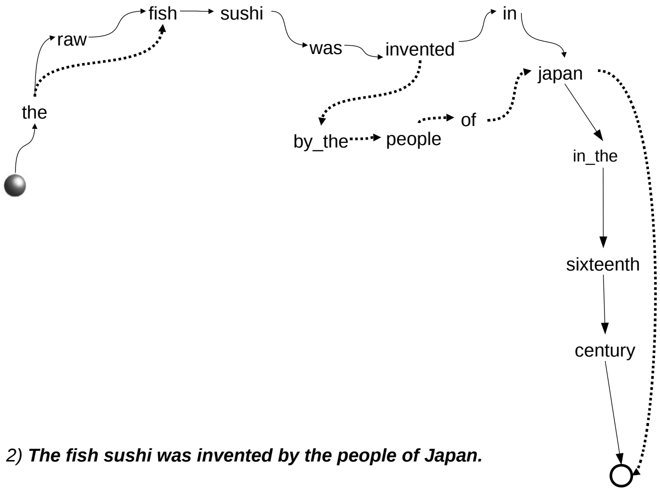


Figure 7.6. *Multisentence fusion: step where sentences 1 and 2 have been added*

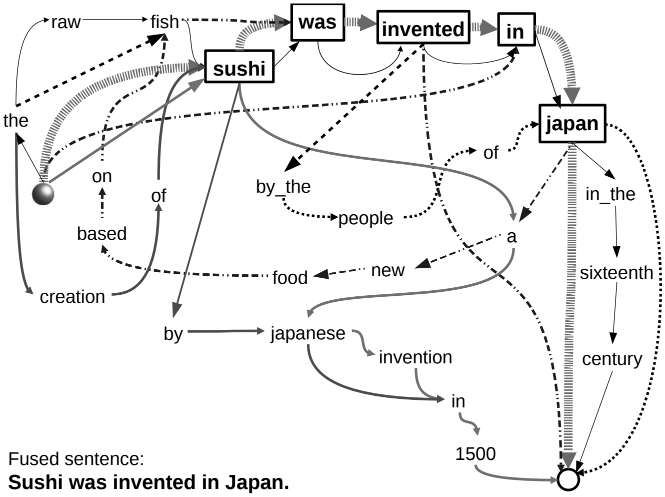


Figure 7.7. *Multisentence fusion: final step where the 5 sentences have been fused*

This method uses very few resources: it only requires parts-of-speech (POS)-tagging and a stoplist of common words. It has been tested in the context of MSF for press articles from GOOGLE NEWS (see section 5.7). The first sentences of articles coming from 150 groups of news in English and 40 in Spanish were, therefore, fused. A manual evaluation measured the informativeness and grammaticality of the fused sentences on a scale of 0– 2 (0 being the worst grade). [FIL 10] reports high rates of grammaticality and informativeness for English documents (see Table 7.3).

Language	Grammaticality = 2	Informativeness = 2
English	64%	52%
Spanish	50%	40%

Table 7.3. *Grammaticality and informativeness scores for the multisentence fusion system by [FIL 10]*

7.6.2. *Graphs and keyphrase extraction: the* TAKAHÉ *system*

The work of Boudin and Morin [BOU 13] built on the research presented in previous section, particularly with regard to three points:

– punctuation was taken into account to produce correctly punctuated and more informative compressions;

– automatic machine translation (MT) approaches and classic summarization evaluation metrics were used to evaluate the performance of MSF systems;

– an evaluation corpus in French with manually created reference compressions was introduced[5].

The authors developed the TAKAHÉ system[6] and created a manually annotated corpus. The evaluation criteria concerning grammaticality and informativeness were inspired by the criteria used by [BAR 05] and [FIL 10], on a scale of 0–2:

[5] The dataset is downloadable at: https://github.com/boudinfl/lina-msc.

[6] Available from: https://github.com/boudinfl/takahe.

Grammaticality: [BAR 05]

– Grammatical (2), if the compression does not contain any grammatical mistakes.

– Partially grammatical (1), if the compression contains at most one mistake with regard to articles, tenses or punctuation.

– Not grammatical (0), if the compression is corrupted by multiple mistakes.

Informativeness: [FIL 10]

– Perfect (2), if the compression conveys the gist of the main event and is more or less like the summary the person would produce himself.

– Related (1), if it is related to the main theme but misses something important.

– Unrelated (0), if the summary is not related to the main theme.

Table 7.4 displays the results obtained in grammaticality and informativeness. According to this table of results, the two systems are very similar in terms of performance.

Method	**Grammaticality**			**Informativeness**		
	0	1	2	0	1	2
Filippova	**9.2 %**	18.3 %	**72.5 %**	10.0 %	**46.7 %**	43.3 %
TAKAHÉ	11.7 %	**23.3 %**	65.0 %	**2.5 %**	35.0 %	**62.5 %**

Table 7.4. TAKAHÉ *scores for grammaticality and informativeness [BOU 13]*

Tzouridis *et al.*'s very recent method [TZO 14] builds a shortest path algorithm set, based on a generalized linear model in a common space of word graphs and compressions. They use an SVM approach to adapt the parameters of the optimization. Tzouridis *et al.* [TZO 14] approach outperforms both the methods presented here. Indeed, the authors observe significant improvements of about 7% in ROUGE and 8% in BLEU score on the MSF task.

7.7. Sentence compression

Automatic sentence compression is a recent topic of research. The basic idea consists of eliminating non-essential information from the sentences in a document, while maintaining their grammaticality; this can be very difficult to implement. The applications for sentence compression are very diverse. To name just a few:

– Automatic title generation. News agencies receive large volumes of information from heterogeneous sources on a daily basis. These organizations have specialists whose job is to give a title to each piece of information (upon arrival), which is then converted into news. [WIT 99] present a system that is capable of generating headings of arbitrary length.

– Subtitle generation for media. Nowadays, most films are subtitled, though most television channels continue to offer a limited subtitling service. However, over recent years, this question has sparked a considerable interest. On the one hand, subtitles can translate a story or a dialog spoken in a foreign language and, on the other, they act as an aid for disabled people: [GRE 98] presents a text reduction method that aims to reduce the reading time of a synthesizer for the visually impaired.

– Applications for mobile phones. Currently, mobile devices have small screens, and the number of characters that can be displayed is limited. Sentence compression is a method that enables the size of the text on display to be reduced and, consequently, more information to be included in a given space.

– Optimization of automatic text summarization systems by further compressing the sentences extracted by text summarization systems [KNI 02, MOL 10b]. A summary generated by sentence compression is a text in which the sentences may have been compressed, though never deleted [YOU 07].

We are going to intuitively define the task of sentence compression as follows:

DEFINITION 7.2.– *Given an input sentence s, a system must produce a sentence as output s_o:*

– that is shorter than the original: $|s_o| \leq |s|$;

– that retains the important information from the original s;

– that is grammatically correct.

Moreover, we are going to borrow [YOU 07]'s formal definition of a compressed sentence:

DEFINITION 7.3.– *Let* $s = (w_1, \ldots, w_i, \ldots, w_n)$ *and* $s^* = (w_1, \ldots, w_j, \ldots, w_k)$ *two sentences, defined as a sequence of words, with* w_i *the* i*th word in the sentence* s*,* w_j *the* j*th word in the sentence* s^* *and* $n > k$*.* s^* *is a compression of* s *if and only if* $s^* \subseteq s$*.*

In general, automatic text summarization systems follow the paradigm of extraction [LUH 58, EDM 69, LAL 02] to locate the most relevant sentences in a text. The task of abstraction-based automatic summarization from extracted text is extremely complex because the summary must include the most relevant content from the original text and write it in a different way [ONO 94, PAI 90]. In some tasks, the summary has a very limited size (such as news summaries). Sentence compression, therefore, enables larger volumes of information to be included in the same space, while preserving grammaticality.

Compression is, therefore, used as a resource for optimizing automatic text summarization systems. Lin's work [LIN 03a] is a direct precursor to this optimization. In his work, sentences from an extraction-based multidocument summarization system were compressed. Indeed, sentence compression is a step toward abstraction, i.e. a primary form of paraphrasing.

The website for the RPM2 project[7] contains important resources for compressing sentences in French. In particular, a *corpus* of 8,388 pairs {original sentence–compressed version} was created from journalistic documents [LOU 10]. Each sentence was manually compressed by four annotators. Annotators were instructed to keep two major elements for good sentence compression: grammaticality and conciseness. The annotators compressed approximately 100 sentences an hour. They eliminated words or groups of words. After compression, they made sure that no word had been modified (change of tense, elimination of apostrophes etc.). This *corpus*, under General Public License (GPL) license, is available from: http://rpm2.talne.eu.

[7] http://rpm2.org/.

Table 7.5 shows a translated example from this *corpus*. Four annotators compressed source sentence number 461. All four annotators eliminated "*While waiting*", but only annotators 1 and 3 obtained the same compression. In longer sentences, discrepancy was often much larger.

Source	While waiting, he should be actively campaigning for his future election to the Académie française...
461-1	*he should be campaigning for his election to the Académie française...*
461-2	*he should be actively campaigning for his future election to the Académie française...*
461-3	*he should be campaigning for his election to the Académie française...*
461-4	*he should be campaigning for his future election to the Académie française...*

Table 7.5. *Example of manual sentence compression (sentence 461, RPM2 corpus) realized by four annotators*

Automatic sentence compression has been tackled using two methods: symbolic approaches and statistical approaches. A brief overview of this subject will be presented as well as a hybrid method that uses linguistics and statistical methods.

7.7.1. *Symbolic approaches*

With regard to symbolic approaches, Cordeiro *et al.*'s work [COR 09] is very important. He developed an unsupervised system that starts by identifying similar sentences with a high probability of being paraphrased in journalistic articles written in English from the web. Next, these sentences are aligned and processed by inductive logic programming (ILP) so that a set of first-order logic predicates can be obtained. These predicates constitute the compression rules. [JIN 00a] describes a complex system that involves both the verification of coherence via the syntactic analysis of sentences and contextual information obtained using WordNet [8].

[YOU 06, YOU 07] present a method based on transformation rules that are applied to syntactic trees for French sentences. Gagnon and Sylva [GAG 06] describe a symbolic method for simplifying French sentences based on syntactic analysis. This method seeks to reduce the length of the sentence by cutting certain non-essential components; sentences are analyzed by a parser that identifies the syntactic dependency and grammatical relations between the words. The output is a tree representation. Sentences are simplified by cutting out the subtrees that have been identified as not being essential to the sentence structure. The main proposition is preserved, making simplifying sentences less than six words long impossible. An evaluation shows that 80%–90% of sentences can be reduced. Of these, approximately 25% of reductions are judged to be incorrect by an evaluator. This rate, which depends a great deal on the quality of the syntactic analysis, is perhaps too high for automatic summarization.

Cohn and Lapata [COH 09] use syntactic dependency trees and transformations enabling both nodes and the structure of the tree itself to be modified. In this way, the most aggressive compressions can be realized while paying attention to grammaticality. Their results show a compression rate of 67%–82%, slightly lower performing than those of the references (57%–76%). Manual evaluations, however, gave a better grade to their system than to other automatic approaches.

7.7.2. *Statistical approaches*

With regard to the line of statistical approaches, works by [KNI 00, KNI 02, MAR 00b] are considered the pillars of research into statistical compression. The authors opt for the *Noisy Channel* model, which is widely used in the domain of statistical MT. Although these studies were conducted for English, in theory the methodology is general enough to be applied to other languages. Lin [LIN 03a]

[8] See section 6.3.1 concerning WordNet and EuroWordNet.

confirms that this method could be of interest for the task of automatic summarization. [MOL 10a] studied sentence compression for the optimization of automatic summarization and their results confirm some of [LIN 03a]'s observations.

Subsequently, Hori and Furui [HOR 04] showed that compression is also useful for speech summarization. Turner and Charniak [TUR 05] show certain problems linked to the *Noisy Channel* model, for example, that it only compresses sentences a small amount. Similarly, Clarke and Lapata [CLA 06b] point out that in this model, compression depends on the domain of the learning *corpus*. [CLA 06a] present an unsupervised model that tackles the task of compression as a problem of linear programming. Recently, [FER 09b] and [WAS 08] showed interesting results with different methods based on statistical physics applied to documents in French and English.

The noisy channel model

In 2002, Knight and Marcu [KNI 02] presented two methods for generating compressed sentences: the noisy channel model, from the domain of MT (see section 6.6.2), and the decision tree learning model. They used the English Ziff-Davis corpus to test their compression approaches. To create this corpus, pairs {sentence/ compressed-sentence} were automatically extracted from a set of articles and their summaries. A pair was added to the corpus if the compressed sentence (from the summary) was a subsequence of a sentence in the article. One thousand and sixty-seven pairs were extracted in this way.

In sentence compression, the noisy channel model is defined as follows: let us take a short sentence s, where via a communication channel other information (noise) is added to the sentence s to generate a long sentence t. Therefore, from the long sentence t, it is a question of finding the most probable sentence of origin s. Let $p(s)$ be the model of the source (*a priori* probability to produce the short sentence) and $p(t|s)$ the model of the noisy channel (probability of transforming the sentence s into t). Indeed, the best compression (a decoder) is, therefore, found by maximizing:

$$\arg\max_{s} p(s) \times p(t|s) \qquad [7.1]$$

According to Knight and Marcu [KNI 02], words can be added, but they cannot be permuted or deleted. Consequently, in this configuration, set C of possible compressions is the set of sentences that can be taken from t by deleting no or several words. Given a sentence of n words, there are 2^n possible compressions (each word can either be conserved or deleted). This model was built on by Cohn and Lapata [COH 09] to take into account word insertions, substitutions and rearrangement.

The noisy channel model requires that $p(t|s)$ be estimated. Knight and Marcu [KNI 02] calculated the probability of the set of operations that would be necessary to convert the syntactic tree of s in the syntactic tree of t. The probability of these operations is estimated using the frequency of these operations in the Penn Treebank corpus [9]. Two major problems linked to learning rules with context-free grammars (CFGs) are: (1) it is impossible to derive all the trees of compressed sentences with CFG productions and (2) several rules are only seen once in the *corpus*.

Knight and Marcu's results [KNI 02] on a corpus of 32 test sentences showed that the noisy channel model is "closer" to human compression than a baseline, which would only eliminate the most frequent bigrams. However, proximity in relation to human compression is relative. A study concluded that the lack of examples in the Ziff-Davis corpus caused the poor quality of the generated compressions [GAL 07].

7.7.3. *A statistical-linguistic approach*

This approach to sentence compression combines discursive segmentation (based on rhetorical structure theory, see section 3.10), calculating textual energy (see section 3.9) and language models. There are two objectives: to determine which chunks of the sentence

[9] http://www.cis.upenn.edu/treebank/.

should be deleted while maintaining the grammaticality of the sentence [MOL 13c]. As has been previously indicated, deleting isolated words prevents sentences from being compressed further. Approaches based on the noisy channel model do not take the context of the segments into account because they act at the word level. Indeed, eliminating isolated words is not effective because their context must be considered.

The approach introduced by Molina [MOL 13a, MOL 13c] presents a method for sentence compression via textual energy (see section 3.9). Textual energy captures the intrasentence relations between the terms and thereby includes the relations between the sentences. This constitutes an elementary form for determining the context of the sentences.

The idea consists of calculating the energy of the elementary discourse units (EDUs) in addition to the energy of the complete sentence. The less informative an EDU the more likely it is to become a candidate for compression. Therefore, informativeness is approached by textual energy, and the system calculates a list of candidates for deletion.

Figure 7.8 presents the architecture of Molina's hybrid system [MOL 13c] for sentence compression.

In this algorithm, informative content is, therefore, approached by the textual energy of each sentence. In order to maintain the grammaticality of a sentence, a language model based on n-grams was used.

A linear combination was proposed to calculate the probability of eliminating a segment $p_e(\sigma)$:

$$p_e(\sigma) = \lambda_E \mathrm{E}(s,\sigma) + \lambda_G \mathrm{Gramm}(s,\sigma) + \lambda_S \mathrm{Seg}(s,\sigma) + \lambda_L \mathrm{Len}(s,\sigma) \qquad [7.2]$$

for reasons of clarity, the notation σ will be used to represent the set of k EDUs in the sentence s. $\mathrm{E}(s,\sigma)$ is the energy of the sentence s and of its EDUs σ, $\mathrm{Gramm}(s,\sigma)$ is the grammaticality of the sentence s

and of its EDUs σ, $\text{Seg}(s, \sigma)$ is the impact of the segmentation and $\text{Len}(s, \sigma)$ is the length of the sentence and its EDUs. The parameters to adjust regression were learnt on a manually annotated corpus with approximately 64,000 judges decisions about whether to eliminate an EDU or not. The corpus is available from: http://molina.talne.eu/sentence_compression/data.

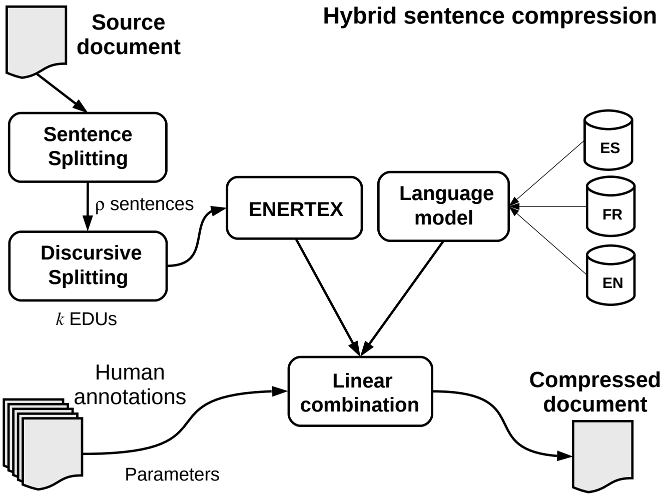


Figure 7.8. *Hybrid system based on textual energy and discourse segmentation*

This system was evaluated with a different approach: the Turing test revisited. Fifty-four judges were asked to guess whether a set of 12 summaries was written by a human or a machine. Of this set, six summaries were randomly taken from a corpus of 2,877 summaries produced by the annotators and six were produced by the aforementioned compression algorithm. For evaluation, the F-test or Fisher test[10] was chosen to interpret the answers given. The null hypothesis H_0 (hypothesis of independence) was fixed to consider that

[10] "An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled" (source: Wikipedia: http://en.wikipedia.org/wiki/F-test).

there was no association between the answers and origin of the summary. The alternative hypothesis H_1, on the other hand, considered there to be a positive association. According to Molina *et al.* [MOL 13b]: “Out of the 54 judges, only one person [...] presented statistical evidence showing that they were able to distinguish between the two types of summary”. It can, therefore, be inferred that 53 judges considered synthetic summaries to be of similar quality to human summaries.

7.8. Conclusion

The automatic production of abstracts is the ultimate aim of research into automatic text summarization. Unfortunately, this objective is still far from being reached. Many obstacles still need to be overcome with regard to the coherence, cohesion and fluency of reading an abstract. Currently, abstraction-based methods are restricted to very limited domains (events such as terrorism or attacks), and the generation rules are still produced manually. That said, abstraction algorithms have made great progress due to IE techniques and NLG.

Pure extraction has started to show its limits, but complementary techniques (numerical and symbolic) are enabling extracts to become increasingly similar to human abstracts. In particular, postprocessing sentence compression is a serious alternative to reformulate extracted sentences. Methods that take the context of EDUs into account and not merely isolated words are promising for the task of sentence compression. With regard to MSF, it consists of generating a single sentence from the fusion of a group of k related sentences. Thus, the main objective is to reduce redundancy, while maintaining the essential information in k sentences. This generated sentence is not necessarily present in the source documents.

Graph-based methods, combining MT approaches, are proving to be very promising for MSF. Compression and fusion methods, therefore, constitute steps toward abstract generation. In any case, it is a matter of urgency that research into automatic text summarization turns more toward developing generative methods for abstract production.

8

Evaluating Document Summaries

Evaluating the quality of summaries produced by automatic systems has always been a subjective and extremely difficult task to implement. The task is subjective because a “perfect” summary does not exist. Indeed, evaluating summaries is an open problem to which the scientific community has responded with partial solutions. Generally, the evaluation of the summaries of documents can be classified into intrinsic evaluation and extrinsic evaluation [SPÄ 96]. Intrinsic evaluation can be conducted with a direct reading – or manual evaluation – of the summary produced. It can also be carried out semi-automatically or automatically. In contrast, extrinsic methods evaluate summaries indirectly while carrying out a specific task [DOR 05], such as combining the results of the summarizer with a question-answering (QA) system [MAN 99a]. In this way, the performance of the QA system can be evaluated: the automatically generated summary enters the system, rather than the whole corpus. This chapter will provide an overview of some techniques used to evaluate automatic summaries. It is worth pointing out that these techniques have been developed in accordance with the performance of summarization systems and tasks specified by the scientific community.

8.1. How can summaries be evaluated?

What generally goes into a summarization system is one (or several) plain text *corpus* expressed in natural language. What comes out of the system are summarized documents also expressed in natural language. Evaluating summaries is, therefore, no easy task. Over the past half-century of research, there have been several attempts to evaluate summaries (see, for instance, [EDM 69, MAT 72, SPÄ 96, LIN 04, MCK 05, SAG 10, LOU 13]), although no general consensus has ever been reached. To help us better understand the problem, let us consider a well-known passage from *The Little Prince* [1], Chapter XII, which is reproduced in Figure 8.1.

The Little Prince – Chapter XII

The next planet was inhabited by a drunkard. This visit was very short, but it plunged the little prince into a deep state of melancholy. "What are you doing there?" he said to the drunkard, whom he found set in silence before a collection of empty bottles and a collection of full bottles. – "I am drinking," responded the drunkard, with a lugubrious air. – "Why are you drinking?" demanded the little prince. – "To forget," replied the drunkard. – "To forget what?" inquired the little prince, who already pitied him. – "To forget that I am ashamed," confessed the drunkard, while lowering his head. – "Ashamed of what?" inquired the little prince, who wanted to help him. – "Ashamed of drinking!" The drunkard finished abruptly, and shut himself up definitively in silence. And the little prince went away perplexed. "Grown-ups are decidedly very, very bizarre," he said to himself, during the next voyage.

Figure 8.1. *Chapter XII of The Little Prince, A. de Saint-Exupéry*

This *corpus* contains 12 sentences and 148 raw words (tokens counted by Linux's `wc` program). Four people produced summaries, which are shown in Figure 8.2 (they are shown as they were written, including original spelling mistakes which are underlined). The summarizers were required to implement a compression rate of 30% of the length of the *corpus* in sentences, which gave summaries of three sentences. The first summary contains 45 words, the second contains 48, the third contains 28 and the fourth contains 50.

In your opinion, which summary provides more information? Which summary out of the four is the most concise, fluent, coherent and cohesive? Finally, which summary is the best?

It is not difficult to see that answering these questions is no simple task. Worse still, it is utopian to hope to achieve a unique and perfect

[1] http://en.wikipedia.org/wiki/The_Little_Prince.

summary because, as [MIN 01] pointed out, “there is no perfect summary which is independent from user demands and types of texts. Rath *et al.* [RAT 61] had already identified a low overlap rate (approximately 25% in their study) in summaries produced by humans. So we will leave the reader to judge for himself or herself.

Summarizer 1
The little prince visited a planet inhabited by a drunkard. After several bizarre answers from the drunkard, the little prince understood that the drunkard drank to forget that he was ashamed of drinking. The visit was very short and the little prince went away perplexed.

Summarizer 2
The little prince went to a planet inhabited by a drunkard and asked him what he was doing there? The drunkard responded that he drank to forget that he was ashamed of drinking. The little prince went away, perplexed and melancholic, saying to himself “grown-ups are bizarre”.

Summarizer 3
The little prince visited a planet inhabited by a drunkard. He talked to the drunkard and asked him why he drank. The response he received left him perplexed.

Summarizer 4
The following encounter, with a drunkard, plunged the little prince into a state of altruism marked with sadness... “I drink because being **sobre** **remind** *me of how* **ashmed** *I am of being an* **alcoolic***,” explained the man. “Grown-ups are* **decidedley** *very, very* **bizare***,” the little prince said to himself.*

Figure 8.2. *Abstracts of Chapter XII of The Little Prince, produced by four humans*

8.2. Extrinsic evaluations

While some people consider the production of a summary itself to be an elegant task, potential users of summaries are often more pragmatic: they believe that the value of a summary ought to be measured according to its usefulness in carrying out other tasks. To this end, several researchers have attempted to carry out this type of extrinsic evaluation based on tasks related to summarization. The early results were quite encouraging. McKeown *et al.* [MCK 05] compared the performance of people who had analyzed information from a heterogeneous set of documents. Four groups of people received either the original documents, titles, automatically generated summaries or summaries produced by other people. The study showed that people who had been given summaries gave better answers than people who did not have summaries. What is more is that the higher the quality of the summary, the more satisfactory the person's analysis. Dorr *et al.* [DOR 05] conducted a study which asked the following question: are users quicker to make judgments about the information contained in summaries than in the information contained in the full document? They found that users made judgments 65% quicker about the summary than about the original document. However, extrinsic evaluations may not be, strictly speaking, necessary: they have first and foremost a practical use and cannot therefore easily be generalized, as they depend too much on one specific task [GIL 11]. We will henceforth only be looking at intrinsic evaluations.

8.3. Intrinsic evaluations

In intrinsic evaluation, summaries can be evaluated manually, semi-automatically or automatically. Each of these methods has advantages and disadvantages. The first method requires large amounts of human time because each summary must be read, evaluated and validated by a certain number of people (judges). The readability, complexity of language and the presence of major concepts are improved because the summary is carefully evaluated. However, manual evaluation is subjective because there can be considerable differences between judges. To level out this subjectivity, semi-automatic evaluation methods have been developed. These methods calculate the similarity measures between a candidate summary and one or several reference summaries. The term "reference summary" is semantically linked to its equivalent Gold standard. According to Teufel and Monens [TEU 97], a Gold standard is defined as: "We will call the criteria of what constitutes success the gold standard, and the set of sentences that fulfill these criteria the gold standard sentences. Apart from evaluation, a gold standard is also needed for supervised learning". A Gold standard sentence is a sentence from the source text which is aligned with a sentence from the summary on the basis of semantic and syntactic similarity [KUP 95].

Intrinsic methods can be reproduced, but the need for reference summaries produced by people presents a limitation. Given that there is no such thing as a "perfect" summary, summaries can differ a great deal in terms of content [RAT 61]. Writing this type of document requires an in-depth analysis of the text so that the ideas, style and arguments can be extracted: each person does this in a different way. As a result, two equivalent summaries can be produced with a completely different vocabulary.

The third and most recent approach is automatic intrinsic methods: the summary is evaluated in relation to the source documents rather than in relation to summaries produced by people. Statistical divergence measures (Jensen–Shannon or Kullback–Leibler) applied to the probability distribution are prioritized in this approach. This independence (from human references) means that these methods are starting to emerge and are gradually making their way into the scientific community.

8.3.1. *The baseline summary*

To conduct an intrinsic evaluation of the automatic summaries produced by systems, a special type of summary was set out: i.e. the baseline summary.

DEFINITION 8.1.– *The baseline method carries out an elementary task of summarization. It enables systems to compare their results to the base results. It can be defined in several ways because the baseline depends on the task.*

The following baselines are regularly used to evaluate summaries:

– Random: this involves drawing at random and extracting n sentences from the P sentences which constitute the document;

– Lead sentences:extracting the first n sentences in the document;

– Lead-End sentences: this involves extracting the first n and the last m sentences in the document;

– Lead words/chars: extracting the first n words (or characters) in the document.

Despite their simplicity, the Lead sentences and Lead-End sentences baselines performed surprisingly well when evaluating journal-type documents. Indeed, in this genre of document, a significant amount of information is condensed at the beginning and at the end of the text [BRA 95, HOV 99, LIN 97]. Therefore, during evaluation, these baselines are very difficult to beat. In order to make comparisons less subjective during international campaigns, candidate summaries for evaluation often cut out a certain number of words or characters (typically between 100 and 250 words from the DUC/TAC evaluations organized by the NIST and 500 words from INEX tasks).

Other types of baselines are defined according to the specificity of the tasks: guided and multi-document summarization (see Chapter 4), multilingual summarization (see Chapter 5) or tweet summarization (see section 6.6.3). For example, a generic summarization system can constitute a baseline (in order to draw comparisons) in a query-oriented summarization system. Indeed, the summarization system MEAD (see section 4.5.5) was chosen as a baseline for the DUC evaluations.

8.4. TIPSTER SUMMAC evaluation campaigns

In 1998, the TIPSTER SUMMAC[2] (*Summarization Conference*) automatic text summarization evaluations (organized by the National Institute of Standards and Technology (NIST)[3]) were held in Maryland (USA). SUMMAC was the first large-scale, independent of the developer, evaluation of automatic text summarization systems [MAN 02]. Based on activities, which were generally carried out by

[2] http://www-nlpir.nist.gov/related_projects/tipster_summac.

[3] http://www-nlpir.nist.gov/projects/duc.

data analysts from the US government, three broad evaluation tasks were specified:

1) the *ad hoc* task (intrinsic);

2) the categorization task (intrinsic);

3) the question-answering (QA) task (extrinsic).

8.4.1. *Ad hoc task*

The *ad hoc* task focused on indicative summaries, which were adapted to process a specific topic:

– 1,000 press articles for each of the five categories;

– the systems generate a guided summary of each text;

– the evaluators decide whether the summary is relevant or not in relation to the query;

– the evaluators measure recall and precision[4]: what are summaries' degree of relevances in relation to queries? (other measures were also calculated).

8.4.2. *Categorization task*

In the categorization task, the evaluation sought to find out whether a generic summary could really present enough information to allow an analyst to quickly and correctly categorize a document:

– 1,000 press articles for each of the five categories;

– the systems generate a generic summary of each text;

– people classify the summaries into five categories;

– the evaluators measure recall, precision and F-measure, among others: how are summaries classified in relation to the full texts?

[4] Recall, precision and F-measure are three traditional measures of relevance in information retrieval (see Appendix A.3).

8.4.3. *Question-answering task*

This task was an extrinsic evaluation: a topic-guided summary was evaluated in terms of its "informativeness" in order to find out the extent to which answers to questions are found in summaries:

– the testers created several questions for each category;

– the systems created summaries without knowing the questions;

– people tried to answer the questions from the text's source and the summaries;

– the evaluators measured the recall of the questions: how many questions can be answered correctly from summaries? (other measures were also calculated).

SUMMAC organized an evaluation of automatic text summarization which proved to be very effective for measuring relevance. Summarization algorithms enabled text to be eliminated (83% and 90% for guided and generic summarization, respectively), while obtaining the same level of relevance as the source documents and reducing analysis time by about half. The QA task presented new automatic methods for measuring the informativeness of precise topic-guided summarization. The content-based automatic scoring correlates positively with the scores produced by the human judges. The evaluation methods used in the SUMMAC campaign had two interests: evaluating summaries and evaluating other results connected to natural language processing (NLP) technologies.

8.5. NTCIR evaluation campaigns

The third NTCIR workshop (2001-2002) centered on evaluating information extraction systems (IR task), question-answering systems (QA task) and automatic text summarization systems (automatic text summarization task). Organized by the NTCIR in Japan, the Text Summarization Challenge 2 (TSC)[5] focused on automatically

[5] http://research.nii.ac.jp/ntcir/ntcir-ws3/work-en.html.

summarizing texts published from 1998 to 1999 in the Japanese newspaper *Mainichi*[6]. There were three objectives:

– to promote research into IR, QA and text summarization by providing reusable test *corpora*;

– to create an exchange forum for research groups interested in comparing results and ideas in an informal atmosphere;

– to improve the quality of test *corpora* based on participants' comments.

8.6. DUC/TAC evaluation campaigns

Since 2001, the NIST has organized the *Document Understanding Conference* (DUC)[7] campaigns to evaluate the performance of NLP algorithms. From 2008, these campaigns were renamed *Text Analysis Conference* (TAC)[8]. The TAC campaigns are much more ambitious than the DUC campaigns. From 2008, the TAC organized workshops on four topics: summarization, question and answering, recognizing textual entailment and knowledge base population. The objective of the DUC/TAC campaigns is twofold: first, to promote progress in the domain of automatic text summarization and, second, to enable researchers to participate in large-scale experiments, which enable them to both develop and evaluate their systems.

The DUC/TAC campaigns successively introduced the following tasks:

– DUC 2001-02: single and multi-document generic summarization (from newswire/newspaper documents);

– DUC 2003: headlines and short (and very short) multi-document summarization;

– DUC 2004; short multilingual summaries, multi-document and biographic summarization;

[6] See http://research.nii.ac.jp/ntcir/ntcir-ws3/tsc/cfp-en.html for more details.

[7] http://duc.nist.gov.

[8] http://www.nist.gov/tac

– DUC 2005: guided summarization and biographic summarization guided by questions on the topic (question-focused summarization);

– DUC 2006: guided multi-document summarization ("question-focused summarization task that required summarizers to piece together information from multiple documents to answer a question or set of questions");

– DUC 2007: guided multi-document summarization for lengths of or up to 250 words, from clusters of about 25 documents; update summarization (Pilot task);

– TAC 2008: update summarization task and opinion summarization (Pilot task OS);

– TAC 2009: update summarization task and Automatically Evaluating Summaries of Peers (AESOP) task;

– TAC 2010: guided multi-document summarization and AESOP task;

– TAC 2011: guided multi-document summarization, AESOP, and Multiling Pilot task, to prioritize and promote the use of multilingual algorithms for summarization. This involves transforming an algorithm or a set of monolingual resources into a multilingual version[9];

– TAC 2014: The Summarization track will focus on contextualization and summarization of biomedical articles (BiomedSumm Track)[10].

Throughout these campaigns, the "topic" guiding the production of multi-document summaries comprised a title as well as a set of questions, both written in natural language.

8.6.1. *Manual evaluations*

Since 2005, the form and the content of summaries produced by participating systems have been evaluated intrinsically and manually. For form, the evaluators assess summaries according to a qualitative

[9] The ACL Multi-documents Multiling task (see section 5.2) replaced the TAC MultiLing Pilot from 2013.

[10] See the TAC'14 Website: http://www.nist.gov/tac/2014/BiomedSumm/index.html.

five-point scale based on the following linguistic criteria (established by the NIST [11]):

– *Grammaticality*: the summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g. fragments and missing components) that make the text difficult to read.

– *Non-redundancy*.: there should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g. “Bill Clinton” when a pronoun (“he”) would suffice.

– *Referential clarity*: it should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

– *Focus*: the summary should have a focus; sentences should only contain information that is related to the rest of the summary.

– *Structure and Coherence*: the summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body information about a topic.

A grade (between 1 and 5) is given for each linguistic criterion: 1 Very Poor, 2 Poor, 3 Barely Acceptable, 4 Good and 5 Very Good. Grade 5 is equivalent to a summary produced by a human.

For the evaluation of content, the judges give two grades for automatic summaries which always use the same grading scale:

– Content responsiveness: “Responsiveness should be measured primarily in terms of the amount of information in the summary that actually helps to satisfy the information need expressed in the topic statement. The linguistic quality of the summary should play only an indirect role in your judgment, insofar as poor linguistic quality

[11] See the webpage: http://duc.nist.gov/duc2005/quality-questions.txt.

interferes with the expression of information and reduces the amount of information that is conveyed".

– Overall content quality: "How relatively well each summary responds to the topic?"

Manual evaluations are vital and have been the key to the success of the DUC/TAC campaigns. They are a good compromise for judging both the form and the content of summaries.

8.7. CLEF-INEX evaluation campaigns

Initiative for the Evaluation of XML retrieval (INEX) has organized campaigns to evaluate the performance of algorithms for auestion-answering (2011) and tweet contextualization (2012–2014) tasks[12]. These tasks are oriented at evaluating extraction-based summarization systems as well as the use of Wikipedia resources.

The INEX campaigns successively introduced the following tasks:

– 2011: short summaries which answer a question. "The QA task to be performed by the participating groups of INEX 2011 was contextualizing tweets, i.e. answering questions of the form *'what is this tweet about?'* using a recent cleaned dump of the Wikipedia". Participants were expected to produce a short summary of less than 500 words, which was made up of fragments extracted from the Wikipedia corpus. The proposed task contained 132 topics.

– 2012/2013: tweet contextualization task. The system must provide a small text that gives additional information about the subject of a given tweet. The main objective is to help the readers to understand the tweet. The text takes the form of a readable summary of less than 500 words, generated from the Wikipedia corpus. The test data were 1,000 and 598 tweets in English in 2012 and 2013, respectively. Tweets were selected from institutional accounts, such as @CNN, @TennisTweets, @PeopleMag and @science.

[12] https://inex.mmci.uni-saarland.de/tracks/qa/.

– 2014: tweet contextualization task. This is a variant of CLEF RepLab[13]: the use of a tweet and an entity associated with the tweet. Using a tweet and a related entity, the system must provide a summary of less than 500 words about the subject of the tweet from the perspective of the entity. In INEX's words, this will enable the following: "answering questions of the form *why this tweet concerns the entity?*". 240 tweets (of at least 80 characters) in English have been selected from CLEF RepLab 2013.

Figure 8.3 shows the INEX tweet contextualization task at CLEF 2012.

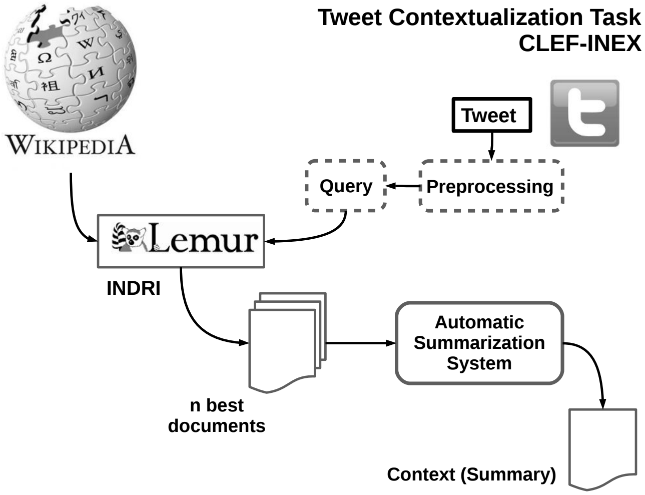


Figure 8.3. *CLEF-INEX tweet contextualization task*

INEX organized two types of evaluation: informativeness and readability. The former is based on content and the latter is based on the fluency of the summary.

The organizers measured informativeness as follows: "[...] the way they overlap with relevant passages (number of them, vocabulary and

[13] http://clef2014.clef-initiative.eu/index.php?page=Pages/labs.html.

bigrams included or missing). For each tweet, all passages from all participants will be merged and displayed to the assessor in alphabetical order. Therefore, each passage's informativeness will be evaluated independently from others, even in the same summary. Assessors will only have to provide a binary judgment on whether the passage is worth appearing in a summary on the topic, or not". Once the important passages have been identified (in other words, the references), the FRESA system, based on the divergence of probability distribution (see section 8.9.2), measured the divergence of candidate summaries in relation to the references and ranked the participating systems. According to the INEX Website, the procedure for establishing readability is as follows:

"Each participant will have to evaluate readability for a pool of summaries on an online web interface. Each summary consists of a set of passages and for each passage, assessors will have to tick four kinds of check boxes:

– *Syntax*: the passage contains a syntactic problem (bad segmentation for example),

– *Anaphora*: the passage contains an unsolved anaphora,

– *Redundancy*: the passage contains a redundant information, i.e. an information that has already been given in a previous passage,

– *Trash*: the passage does not make any sense in its context (i.e. after reading the previous passages). These passages must then be considered as trashed, and readability of following passages must be assessed as if these passages were not present."

8.8. Semi-automatic methods for evaluating summaries

8.8.1. *Level of granularity: the sentence*

Evaluating summaries at the sentence level can be conducted semi-automatically using traditional similarity measures from information retrieval, such as precision and recall (see Appendix A.3) [CHU 00, KUP 95, NET 02]. Evaluators produce a set of so-called "perfect" summaries (reference summaries), one for each document in the test set; they then compare the results of the system with the set of references by measuring content overlap with precision and recall. These measures indicate how close a system's performance can get to manual performance.

With a sentence-level approach, precision measures how many sentences extracted by a system were also retained by people (reference sentences) and recall indicates how many reference sentences were forgotten by a system. The measure F combines precision and recall in a harmonic mean (see Appendix A.3).

Let Sum_{ref} and Sum_{can} be the set of reference sentences selected by a human judge and by a system, respectively. The equations for calculating precision P, recall R and the F-measure are therefore as follows:

$$R = \frac{|\text{Sum}_{\text{ref}} \cap \text{Sum}_{\text{can}}|}{|\text{Sum}_{\text{ref}}|}; \quad P = \frac{|\text{Sum}_{\text{ref}} \cap \text{Sum}_{\text{can}}|}{|\text{Sum}_{\text{can}}|};$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \qquad [8.1]$$

Other semi-automatic measures which have a finer level of granularity, the word level, and which are better adapted to automatic summarization have been created. Currently, the most widespread semi-automatic evaluation methods are ROUGE, PYRAMID and BASIC ELEMENTS. These methods will be discussed below.

8.8.2. *Level of granularity: words*

8.8.2.1. *Recall-oriented understudy for Gisting evaluation*

An automatic summary (candidate) can be evaluated semi-automatically by measuring its similarity in relation to one or several reference summaries. From this perspective, Lin *et al.* [LIN 03b, LIN 04] developed the *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE)[14] measure. The idea is simple: the

[14] ROUGE software is available from: http://berouge.com.

differences between the distribution of words in the candidate summary and the distribution of words in the reference summaries are compared. Figure 8.4 shows a schematic description of ROUGE evaluations, when h humans produced h reference summaries. The references and the candidate summary to be evaluated are split into n-grams to calculate the intersection of n-grams between the references and the candidate.

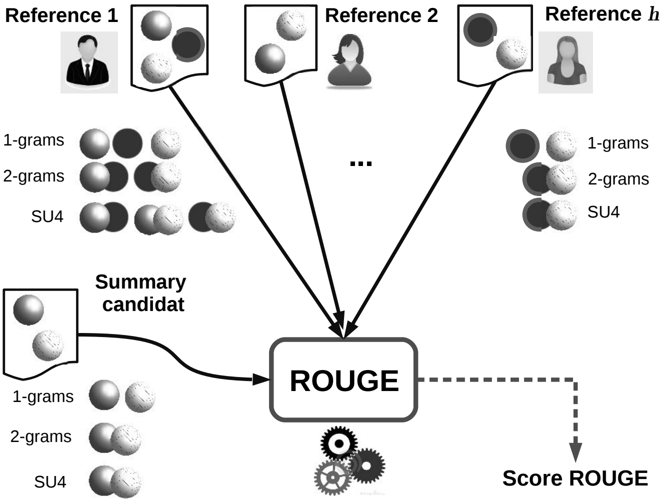


Figure 8.4. *The basic idea of* ROUGE *for the evaluation of summaries*

Widely used during the DUC/TAC evaluation campaigns and inspired by the BLEU[15] measures used in automatic translation, ROUGE is increasingly considered to be standard, given its correlation with manual judgments. In fact, [LIN 04] reports Pearson coefficients for ROUGE-2 and -SU4 at a value of between 0.94 and 0.99.

[15] *BiLingual Evaluation Understudy* (BLEU) (see section 5.9.1.2) [PAP 02]. The ROUGE measure was also used for the evaluation of automatic translation. For these applications, consult Lin C.Y., Och F.J., “Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics”, *ACL’04*, Barcelona, pp. 605–612, 21–26, 2004; and Lin C.Y., Och F.J., “ORANGE: a method for evaluating automatic evaluation metrics for machine translation”, *COLING’04*, Geneva, pp. 501–507, 2004.

Two variations of ROUGE commonly used in the literature are ROUGE-2 and ROUGE-SU4. ROUGE-n: the recall measure is calculated from the co-occurrences[16] of n-grams[17] between a candidate summary Sum_{can} and a set of reference summaries Sum_{ref} (equation [8.2]):

$$\text{ROUGE-}n = \frac{\sum_{n\text{-grams}} \in \{\text{Sum}_{\text{can}} \cap \text{Sum}_{\text{ref}}\}}{\sum_{n\text{-grams}} \in \text{Sum}_{\text{ref}}} \qquad [8.2]$$

where the numerator corresponds to the maximum number of co-occurrences of n-grams in Sum_{can} and Sum_{ref}. The denominator of the equation is the total sum of the number of n-grams present in the reference summaries Sum_{ref}. The ROUGE measure with bigrams, when $n = 2$, is very popular during evaluations.

ROUGE-SUγ: adaptation of ROUGE-2 using *skip units* (SU) of a size $\leq \gamma$. SU4 considers the bigrams and the bigrams SU to be arbitrary in a maximum window length of $\gamma = 4$ words.

The number of bigrams with an arbitrary size of length γ is given by:

$$\text{Count}(k, n) = C\binom{n}{k} - \sum_{0}^{k-\gamma}(k - \gamma) \; ; \; \gamma \geq 1 \qquad [8.3]$$

where n is the n-gram length and k is the sentence length in words.

[16] "Co-occurrence or cooccurrence is a linguistics term that can either mean concurrence/coincidence or, in a more specific sense, the above-chance frequent occurrence of two terms from a text corpus alongside each other in a certain order. Co-occurrence in this linguistic sense can be interpreted as an indicator of semantic proximity or an idiomatic expression. In contrast to collocation, co-occurrence assumes interdependency of the two terms" (source: Wikipedia: http://en.wikipedia.org/wiki/Co-occurrence).

[17] "In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application" (source: Wikipedia: http://en.wikipedia.org/wiki/N-gram).

For instance, the sentence “Intellectuals solve problems; geniuses prevent them”[18] has 6 unigrams, 5 bigrams and Count$(6, 2) = 12$ bigrams SU4, which are shown in Table 8.1.

Sentence	***Intellectuals solve problems; geniuses prevent them.***	**Count**
ROUGE-1	Intellectuals ▷ solve ▷ problems ▷ geniuses ▷ prevent ▷ them	6
ROUGE-2	Intellectuals solve ▷ solve problems ▷ problems geniuses ▷ geniuses prevent ▷ prevent them ▷	5
ROUGE-SU4	Intellectuals solve ▷ Intellectuals problems ▷ Intellectuals geniuses ▷ solve problems ▷ solve geniuses ▷ solve prevent ▷ problems geniuses ▷ problems prevent ▷ problems them ▷ geniuses prevent ▷ geniuses them ▷ prevent them	12

Table 8.1. *n-grams generated by several* ROUGE *measures*

Despite the exhaustiveness of the n-grams, ROUGE has limits relating to the representation of content. For example, the co-occurrence of n-grams is at a superficial level and, therefore, chains of words such as:

– “Great white shark” $\neq$ “White death”;

– “The airplane A380” $\neq$ “The Airbus 380 aeroplane”;

– “ATS” $\neq$ “Automatic Text Summarization”;

have few or no co-occurrences in ROUGE.

Does ROUGE’s calculation of content have other limits? Indeed, the following question must be asked: it is possible to cheat when evaluating text summaries and obtain very high ROUGE scores? Unfortunately, the answer is yes. Sjöbergh [SJÖ 07] conducted a study in 2007 which showed that a simple algorithm – without knowledge and without sentence weighting – based exclusively on a Markov automata enables summaries to be generated without extraction, but with a high number of “relevant” bigrams. The summarizer first calculates all the frequencies of bigrams in the source text. The most

[18] A. Einstein.

frequent bigram is selected to start the summary. Then the basic strategy involves selecting the next word in such a way that the new bigram created in the summary is the one with the highest frequency given by the last word in the summary. Figure 8.5 shows a fragment of the "summary" generated by this algorithm. It obtains a ROUGE-2 value of 12.0, while the average for people at DUC'04 was only 11.3.

> of the hurricane andrew had been injured and the storm caused by the gulf of mexico and louisiana and at least 150 000 people died on the us insurers expect to the florida and there are likely to be concentrated among other insurers have been badly damaged a result of damage caused in the state and to dollars 20bn of new orleans then to pay out on monday and new iberia associated tornadoes devastated inhabitants of miami
>
> ⋮
>
> industry is the costliest disaster in florida as a quarter of hurricane hugo which hit the industry analysts cautioned that the bahamas on tuesday night or shut down

Figure 8.5. *Fragment of a summary generated by a "cheat"* ROUGE *algorithm, according to [SJÖ 07]*

8.8.2.2. *Basic Elements*

Strictly in terms of the content of summaries (and not their readability), the ROUGE evaluation suffers from countless lacuna relating to its dependence on n-grams, which are used to calculate scores. Compound words (for instance, "United Mexican States", "2001: a Space Odyssey" and "Université d'Avignon et des Pays de Vaucluse") and meaningful stopwords (articles, conjunctions, pronouns, etc.) can bias the number of co-occurrences and therefore the overall evaluation.

Two methods have been developed to resolve this problem: BASIC ELEMENTS (BE) and PYRAMID.

The BASIC ELEMENTS [HOV 06] approach seeks to incorporate conceptual elements into the calculation of co-occurrences. The BASIC ELEMENTS are broken down into triplet elements (H | M | R), where H = Head, M = Modifier and R = Relation which links the Hs and the

Ms. Through the lexicon, semantics (based on synonymy) and the distribution of identities, BE attempts to make approximate matches at the level of paraphrases or semantic generalizations. Several resources have been used to construct BEs: Charniak's parser [19], a dependency analyzer, a syntactic-unit chunker, segmentation rules, etc. The correlation between BE and ROUGE (section 8.8.2.1) and between BE and Responsiveness is high. They were established on the basis of Spearman coefficients (higher than 0.926 for Responsiveness and 0.882 for ROUGE) and Pearson coefficients (higher than 0.975 for Responsiveness and 0.897 for ROUGE) [HOV 06].

Recently, Tratz and Hovy [TRA 08] advanced the BEs using transformations. For the correspondences of non-identical units with the same meaning, they applied transformation rules to each unit with a certain number of different variants. Basic elements with transformations for evaluation (BEwT-E) [20] software is a package which automatically creates BEs from a document, applies transformation rules to expand these units into countless variants and then matches these units to a list of units produced by BEwT-E from another document.

8.8.2.3. PYRAMID

The third semi-automatic method, PYRAMID [NEN 04, NEN 07], involves extracting the longest elements (such as noun phrases or summarization content units (SCU)) from the reference summaries [21]. According to Nenkova and Passonneau [NEN 04]:

DEFINITION 8.2.– *An SCU consists of a set of contributors that, in their sentential contexts, express the same semantic content. An SCU has a unique index, a weight, and a natural language label.*

Each element is then considered to be a semantic unit (expressing a unique notion) and then weighted according to the number of reference

[19] ftp://ftp.cs.brown.edu/pub/nlparser/.

[20] Available for downloadat http://www.isi.edu/publications/licensed-sw/BE/index.html.

[21] PYRAMID software is available at http://www.cs.columbia.edu/ani/DUC2005/.

summaries containing it. The identification of similar fragments via summaries is counted. For instance:

1) *13 sailors have been killed* $\sim$ *rebels killed 13 people*.

2) *10.1 millions of euros* $\sim$ *near 10 million of euros*.

A pyramid of several SCUs of weight n is constructed from their weighting for n reference summaries (Gold standard summaries). Each SCU at the level T_i of the pyramid has the weight i; the most frequently included elements are thus located at the top. Evaluating a summary therefore consists of comparing the elements it contains. The best summary is one that contains all the units at level n, then all those at level $n - 1$, and so on and so forth. If d_i is the number of SCUs in a summary at level T_i, the weight D of the summary is given by equation [8.4]:

$$D = \sum_{i=1}^{n} i \cdot d_i \qquad [8.4]$$

A high value indicates a maximized coverage in relation to reference summaries.

Figure 8.6 shows a schematic description of SCU from reference summaries which are assembled into a pyramid with three weight levels. The SCUs, which are probably located in the candidate summary, are counted according to their level in a (simplified) pyramid score:

$$\textsc{Pyramid} = \frac{\sum \text{weight}(\text{SCU}_{\text{candidate}})}{\sum \text{weight}(\text{SCU}_{\text{references}})} \qquad [8.5]$$

The problem with BASIC ELEMENTS and PYRAMID methods is that they are difficult to make completely automatic. Indeed, extracting the various sized elements necessarily involves several similar ways of showing the same information, which is not always evident.

8.9. Automatic evaluation via information theory

One of the first works to mention the use of measures based on content was [DON 00]. The authors of this work presented an evaluation method that compared the rankings of automatic summarization systems by using the calculations of recall and cosine similarity between the text and the generated summary. The latter calculation is clearly a measure based on content. They showed that there was a weak correlation between the rankings produced by the calculation of recall, but that measures based on content produced strongly correlated rankings. This opened the way to measures based on content by comparing the content of the automatic summary with the content of the reference summaries. Saggion *et al.* [SAG 02a] presented a set of evaluation measures based on the notion of vocabulary overlap, which includes n-grams, cosine similarity and the longest common sub-sequence. These measures were applied to multi-document summarization in English and Chinese. However, they did not evaluate the performance of these measures on different summarization tasks. Radev *et al.* [RAD 03] also compared different evaluation measures based on vocabulary overlap. Although these measures enable random systems to be separated from non-random systems, no clear conclusion about the relevance of the measures studied was reached.

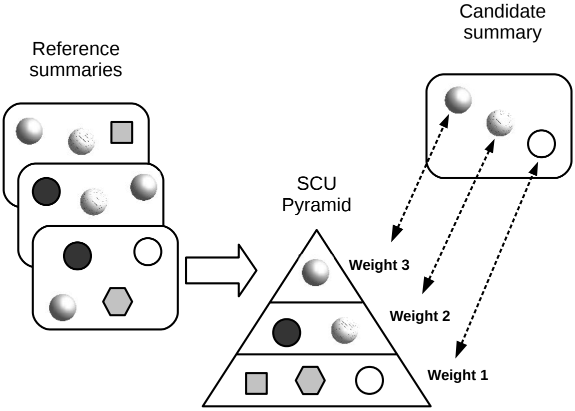


Figure 8.6. *An example of an SCU pyramid*

Several measures at the word level of granularity, which do not require human references to evaluate summaries, will now be presented. These measures are based on the idea of calculating the divergence of the distribution of words between the source and the candidate summary.

8.9.1. *Divergence of probability distribution*

Lin *et al.* [LIN 06] developed an evaluation method based on information theory and the use of divergence measures between two probability distributions: the probability distribution of terms in the automatic summary and the probability distribution of terms in the reference summary. They studied two theoretical information measures: Kullback–Leibler divergence ($\mathcal{D}_{\mathcal{KL}}(P||Q)$, see equation [3.12]) [KUL 51] and Jensen–Shannon divergence ($\mathcal{D}_{\mathcal{JS}}(P||Q)$, see equation [3.14]) [LIN 91]. These measures can be applied to the distribution of words w in automatic summaries P_w and in reference summaries Q_w (with an appropriate smoothing of the summary probabilities). The value obtained is used to give a score to the produced summary. The method was evaluated by Lin *et al.* [LIN 06] using the DUC'02 *corpus* for single- and multi-document summarization tasks. The results show that the performance of evaluation by divergence of probability distribution is comparable to ROUGE-1 evaluation for single-document summarization and better for multi-document summarization. The authors suggested testing other sets of data from the DUC campaigns after 2002.

This work was the starting point for research into the relevance of measures which are not based on human references. Louis and Nenkova [LOU 09] suggested directly comparing the distribution of words in complete documents with the distribution of words in automatic summaries in order to work out an evaluation measure based on content. They recorded a strong correlation between the rankings produced with references and those obtained without references. However, this calculation is strictly limited to the unigrams of distributions, after a stemming of the documents. The authors [LOU 09] used the TAC'08 Update Summarization task *corpus* to calculate the measures $\mathcal{D}_{\mathcal{JS}}$ and ROUGE [LIN 04] for candidate summaries. Two rankings of the systems were conducted and then compared to the rankings produced by PYRAMID and Responsiveness.

Spearman correlations were calculated between these rankings. The results are presented in Table 8.2 with their p-value[22]. They confirm that there is a strong correlation between $\mathcal{D}_{\mathcal{JS}}$, PYRAMID [NEN 04] and Responsiveness [OVE 07]. The evaluation system SIMETRIX [LOU 09] can be downloaded from http://homepages.inf.ed.ac.uk/alouis/IEval2.html.

Measure	PYRAMID; p-value	Responsiveness; p-value
ROUGE-2	0.960; $p < 0.005$	0.920; $p < 0.005$
$\mathcal{D}_{\mathcal{JS}}$	0.850; $p < 0.005$	0.740; $p < 0.005$

Table 8.2. ROUGE-2 *and* $\mathcal{D}_{\mathcal{JS}}$ *versus* PYRAMID *and Responsiveness, TAC'08 update summarization task*

Saggion *et al.* [SAG 10] using FRESA went a step further and conducted a multilingual single and multi-document evaluation of summaries, showing that human references are not needed for certain evaluation tasks.

8.9.2. FRESA

Framework for evaluating summaries automatically (FRESA) is an automatic method based on information theory, which evaluates summaries without using human references [TOR 10b]. FRESA is based on the works of Lin *et al.* and Louis and Nenkova [LIN 06, LOU 09] and can evaluate summaries in English, French, Spanish and Catalan. It conducts a standard preprocessing of documents (language identification, filtering and lemmatization) before calculating the probability distributions P and Q. FRESA calculates the divergence $\mathcal{D}(P||Q)$ via unigrams, bigrams, bigrams SU4 and their combinations. Figure 8.7 shows a simplified version of the idea behind FRESA. The summary to be evaluated (candidate) and the document (source) are split into n-grams in order to calculate the intersection between the source and the candidate.

[22] In statistics, the p-value is the smallest level at which the null hypothesis is rejected.

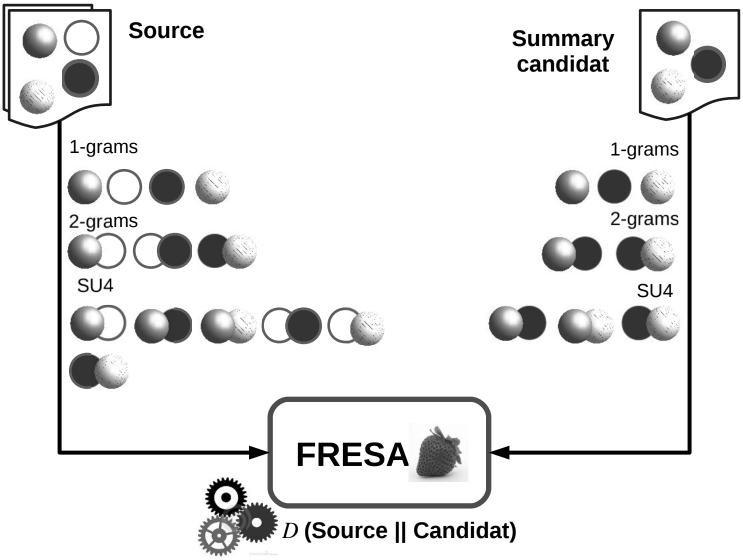


Figure 8.7. *Automatic evaluation of summaries via* FRESA

The authors of [SAG 10] and [TOR 10b] have run large-scale trilingual experiments (seeTable 8.3). For experiments in English, official summaries were used. For experiments in Spanish and French, candidate summaries were created by the systems: ENERTEX [FER 07], CORTEX [TOR 02], SUMMTERM [VIV 10a], REG [TOR 10a], $\mathcal{D}_{\mathcal{JS}}$ summarizer (section 3.6), Lead sentences and random sentence baselines, OTS [YAT 07], *Word*, SSSUMMARIZER, PERTINENCE and COPERNIC SUMMARIZER[23].

The measures ROUGE, Coverage [OVE 07], Responsiveness and PYRAMID were used in the English experiments. The French and Spanish experiments used the measures ROUGE (-1, -2 and -SU4). The measure FRESA was also calculated for each experiment.

The Spearman correlations for DUC'04 are presented in Table 8.4. In task 2, a strong correlation between FRESA and Coverage was recorded. In task 5 (Biographic), the correlation between FRESA and

[23] See Appendix 2 for the websites of these systems.

Coverage is weak and the correlation between FRESA and Responsiveness is weak or even negative. The correlation between FRESA and ROUGE-2 (Spearman correlation = 0.83, not shown in the table) was verified.

Task	*Corpus*	Language	Metrics
Multi-document	DUC’04 (task 2)	English	ROUGE, Coverage, Responsiveness and PYRAMID
Biographic Summarization	DUC’04 (task 5)		
Update Summarization	TAC’08		
Opinion Summarization	TAC’08 (OS)		
Single-document	PISTES	French	ROUGE
Multi-document	RPM2		
Single-document	*Medicina Clínica*	Spanish	ROUGE

Table 8.3. *Corpora used by Saggion et al. and Torres-Moreno et al. [SAG 10, TOR 10b]. RPM2: http://rpm2.org/, PISTES: http://pistes.revues.org and Medicina Clínica: http://zl.elsevier.es/es/revista/medicina-clinica-2*

DUC’04 Task 2	
Measure	**Coverage; p-value**
ROUGE-2	0.790; $p < 0.005$
FRESA	0.680; $p < 0.003$

DUC’04 Task 5		
Measure	**Coverage; p-value**	**Responsiveness; p-value**
ROUGE-2	0.780; $p < 0.001$	0.440; $p < 0.05$
FRESA	0.400; $p < 0.050$	–0.180; $p < 0.25$

Table 8.4. *Measures* FRESA *and* ROUGE *versus Coverage, DUC’04 tasks 2 and 5*

The rankings of FRESA versus the rankings of PYRAMID and Responsiveness in the TAC’08 OS *corpus* were compared: their correlations are shown in Table 8.5. There is a weak, or even negative, correlation between FRESA and PYRAMID or Responsiveness. The correlation between the rankings PYRAMID and Responsiveness is high for this task (Spearman correlation = 0.71).

For the Spanish and French experiments, the results in Table 8.6 show a high correlation between FRESA and ROUGE. The results of these large-scale multilingual experiments show that if we are only interested in ranking summarizers by Informativeness, there are tasks when the source document could substitute the references while obtaining the weak rankings. However, it has been noted that it is not always desirable for the complete document to substitute references, especially in complex summarization tasks such as biographic summarization and opinion summarization.

Measure	**PYRAMID**; p**-value**	**RESPONSIVENESS**; p**-value**
FRESA	-0.130; $p < 0.25$	-0.140; $p < 0.25$

Table 8.5. *Measure* FRESA *versus Responsiveness and* PYRAMID, *TAC'08 OS task*

Corpus	**Measure**	**ROUGE-1**; p**-value**	**ROUGE-2**; p**-value**	**-SU4**; p**-value**
RPM2	FRESA	0.850; $p < 0.002$	0.640; $p < 0.05$	0.740; $p < 0.01$
Medicina Clínica	FRESA	0.820; $p < 0.005$	0.710; $p < 0.02$	0.710; $p < 0.01$
PISTES	FRESA	0.880; $p < 0.010$	0.830; $p < 0.02$	0.830; $p < 0.01$

Table 8.6. FRESA *versus* ROUGE, *corpora multi-document RPM2 (French) and Medicina Clínica (Spanish), and single-document PISTES (French)*

The FRESA system was used to evaluate the quality of summaries of biomedical documents written in Catalan, when the authors' reference summaries were not available [VIV 10b] (see section 6.3.1).

FRESA was also used during the CLEF-INEX campaigns for the question-answering track (QA track, QA@INEX) and tweet contextualization task [SAN 12][24]. QA track consisted of answering an encyclopedic-type question by extracting and agglomerating sentences from Wikipedia. The document/question sets correspond well to the task sets for which high correlation rates between the FRESA values and the semi-automatic evaluation methods were recorded. During these evaluations, FRESA's divergence function was rewritten in order to include the asymmetry of the length of the

[24] FRESA is downloadable at http://fresa.talne.eu.

summary and the source. Indeed, Jensen–Shannon divergence did not seem well adapted to take this ratio into account.

Let D_P and D_Q be a source text and a summary, respectively. The algorithm now implements a modified divergence $\mathcal{KL}$ between the probability distribution P in the source and the probability distribution Q in the summary:

$$\textsc{Fresa}(P||Q) = \sum_{w \in P} P_w \left(1 - \frac{\min(\log_2(P_w \times |P|, \log_2(Q_w \times |P|)}{\max(\log_2(P_w \times |P|, \log_2(Q_w \times |P|)}\right) \qquad [8.6]$$

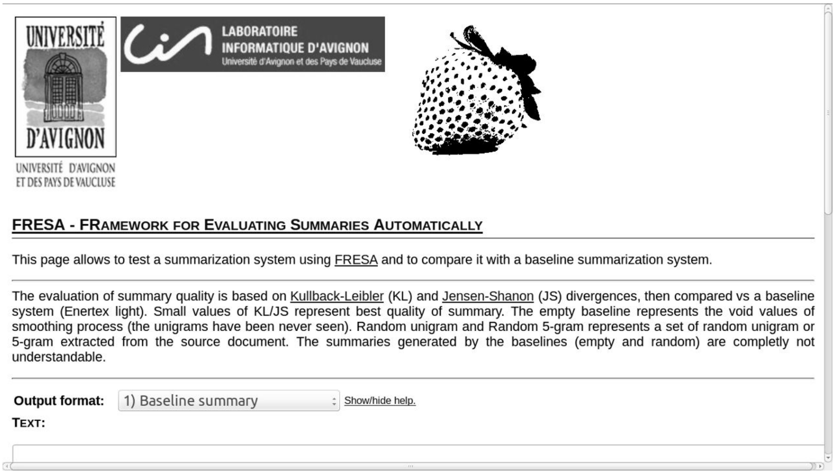
with the following specification for the probability distribution of words w:

$$P_w = C_w^P/|P|; \qquad Q_w = \begin{cases} C_w^Q/|Q| & \text{if } w \in Q \\ 1/|P| & \text{elsewhere} \end{cases} \qquad [8.7]$$

where P_w is the probability distribution of words w in the source text D_P. Q_w is the probability distribution of words w in the summary D_Q. C_w^P is the occurrences of the word w in the source text and C_w^Q is the occurrences of the word w in the summary. $|P|$ is the number of words in the source text and $|Q|$ is the number of words in the summary.

The FRESA user interface is shown in Figure 8.8.

As academic examples, FRESA evaluations of summaries from a chapter of *The Little Prince* (see Figure 8.1) and the full text of this book have been conducted. Table 8.7 shows FRESA's evaluation of the summaries from Chapter XII of *The Little Prince*. Summary number 2 obtained the highest FRESA score = 0.09289, although it is less concise than summary number 3. With regard to the evaluation of the author summary and the summary generated by the CORTEX system of this book (see Figures 1.5 and 1.6, respectively), FRESA obtains a score of 0.00965 for the artificial summary and 0.00593 for the author summary. A random baseline summarizer only obtains a score of 0.00216.


UNIVERSITÉ D'AVIGNON
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE
LABORATOIRE INFORMATIQUE D'AVIGNON
Université d'Avignon et des Pays de Vaucluse

FRESA - FRAMEWORK FOR EVALUATING SUMMARIES AUTOMATICALLY

This page allows to test a summarization system using FRESA and to compare it with a baseline summarization system.

The evaluation of summary quality is based on Kullback-Leibler (KL) and Jensen-Shanon (JS) divergences, then compared vs a baseline system (Enertex light). Small values of KL/JS represent best quality of summary. The empty baseline represents the void values of smoothing process (the unigrams have been never seen). Random unigram and Random 5-gram represents a set of random unigram or 5-gram extracted from the source document. The summaries generated by the baselines (empty and random) are completly not understandable.

Output format: 1) Baseline summary

Figure 8.8. *Interface of the automatic evaluation system* FRESA *at http://fresa.talne.eu/*

8.10. Conclusion

In the previous chapter, three broad approaches to the automatic generation of summaries were presented: abstraction, extraction and compression. Currently, extraction algorithms dominate the landscape and are at the center of countless automatic summarization systems. The ease with which these methods can be implemented and their good performance are the key to their success. This performance must, however, be objectively evaluated. Evaluation enables algorithms to be positioned in relation to a task and an idea about the quality of the summaries to be obtained.

This chapter has described several methods for evaluating summaries (intrinsic, extrinsic, manual and automatic) as well as their difficulties. Manually evaluating summaries (by direct reading) involves a very careful analysis of both the content and the form, but only a limited number of *corpora* can be evaluated. Semi-automatic evaluations are fast, reproducible and can process large volumes of summaries. However, manual evaluation is always costly in terms of time and resources and semi-automatic evaluations require references and are still not very reliable. In fact, no completely automatic evaluation method which has the support of the scientific community exists; however, several semi-automatic partial solutions (such as ROUGE, BASIC ELEMENTS and PYRAMID) do exist, which enable an idea about the quality of the summary to be obtained. What these solutions have in common (which is also their common weak point) is that they require one or several reference summaries. It is worth noting that there is a debate about how many references should be used in the evaluation of summaries [OWK 09]. Evaluation methods based on information theory, using Jensen–Shannon or Kullback–Leibler divergences, appear to give interesting results. The main advantage of these methods is that they constitute a step toward completely automatic algorithms for evaluating summaries without the need of human references. The validity of measures which evaluate the content of summaries without requiring reference summaries has been presented. It is clear that evaluation campaigns have made a great contribution to the success and development of automatic summarization systems and their evaluation. However, these conferences do not escape criticism. As Fort [FUE 08] has pointed out, the main reproach that can be cast at the DUC and TAC campaigns is that their objectives are constantly changing, denying the community of continuity of research into automatic text summarization. Moreover, TAC 2014, in returning to the generation of scientific summaries (a historic task studied by the authors of [LUH 58, EDM 69, RUS 71, MAT 72, KUP 95], among others), shows that even for seemingly well-characterized document genres, summarization is still an open question.

Summary		**FRESA**$_M$
2	The little prince went to a planet inhabited by a drunkard and asked him what he was doing there? The drunkard responded that he drank to forget that he was ashamed of drinking. The little prince went away, perplexed and melancholic, saying to himself "grown-ups are bizarre."	**0.09289**
1	The little prince visited a planet inhabited by a drunkard. After several bizarre answers from the drunkard, the little prince understood that the drunkard drank to forget that he is ashamed of drinking. The visit was very short and the little prince went away perplexed.	0.08626
4	The following encounter, with a drunkard, plunged the little prince into a state of altruism marked with sadness... "I drink because being **sober** **reminds** me of how **ashamed** I am of being an **alcoholic**," explained the man. "Grown-ups are **decidedly** very, very **bizarre**," the little prince said to himself.	0.08338
5	The following encounter, with a drunkard, plunged the little prince into a state of altruism marked with sadness... "I drink because being *sobre* *remind* me of how *ashmed* I am of being an *alcoolic*," explained the man. "Grown-ups are *decidedley* very, very *bizare*," the little prince said to himself.	0.05970
3	The little prince visited a planet inhabited by a drunkard. He talked to the drunkard and asked him why he drank. The response he received left him perplexed.	0.02869

Table 8.7. FRESA*'s evaluation of summaries of* The Little Prince *(Chapter XII, Figure 8.1)*

Conclusion

After more than 50 years of research into automatic text summarization, where are we now? What still needs to be done? In which direction should we go? [1]

This book has discussed automatic text summarization's long journey toward making an artificial system produce abstracts. In automatic text summarization, a system, *in fine*, is confronted with real texts. The text – containing mistakes, contradictions, redundancy and incomplete information – is the concrete object that automatic summarizers must face. Algorithms must generate the most concise and credible summaries possible from a *corpus*. Writing relevant summaries is a very difficult problem for people. For machines, which do not have the faculty to understand the text, the challenge is even greater. Automatic text summarization has, however, come a long way since H.P. Luhn's founding works in 1958 [LUH 58].

Though there is no complete theory for the subject, progress at the application level has been steady. From the very beginning, the first attempt's ultimate aim was to produce abstracts in the same way people produce them, while fixing the main lines of research into automatic summarization. But this objective is still far from being achieved. The difficulty of mimicking an intellectual task which – even

[1] Karen Spärck Jones asked this question back in 1997.

for people – is yet to be mastered was quickly identified. In fact, although the generation of single-document (and monolingual) summaries has not been tackled since the DUC’02 campaign, it remains to this day an open problem [NEN 11]. However, technological advances and research into new algorithms have continued to grow, as has the definition of new tasks, which are making an already complex subject even more difficult. Since the creation of the Internet, the overload of information has contributed a great deal to this dynamic. Users are asking for increasingly higher performing automatic text summarization systems, which produce quality personalized summaries, are able to process large volumes of heterogeneous documents that over time have become multilingual and, moreover, that the task be completed quickly. Fulfilling all these expectations is no mean feat. The basic problems remain deep text understanding and fluent production in natural language.

Several approaches have been developed to try to simulate human performance. To this end, approaches that extract relevant sentences without really being able to understand the text have been preferred over approaches that try to understand documents in order to produce a summary. Extraction is realized using – superficial, intermediate or deep – automatic text analysis methods, which enable relevant sentences to be weighted, concatenated and transformed to generate a summary. Extraction-based methods avoid facing the difficult problem of making a machine understand a text, which is still an unresolved problem. In particular, extractive summarization systems based on statistical methods (such as the vector space model (VSM) or graphs) have given very interesting results. More powerful vector models are starting to be used in automatic summarization. Indeed, recent studies [MAN 09, YIN 12, LIT 14] have modeled multidocument *corpora* in a space beyond that of the VSM: they introduced tensors into multidocument summarizers with topic detection. This idea is shown in Figure C.1.

The documents plunged into a triple dimension – documents-terms-topics – are decomposed via tensors T of order n, defined as a multidimensional matrix of dimension n. The performance is very promising. These models can, moreover, be easily realized by software.

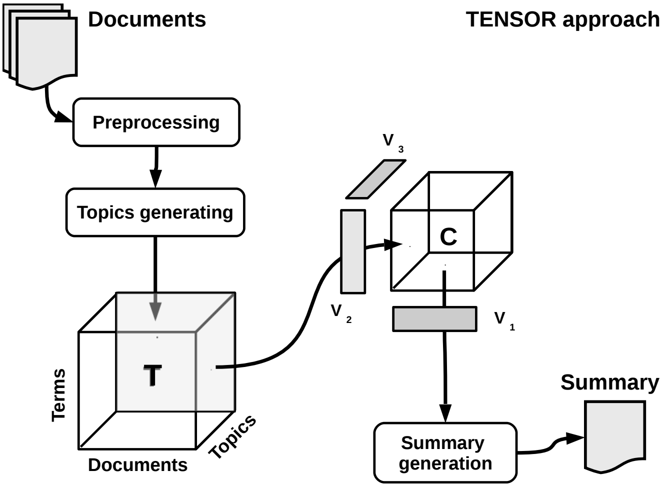


Figure C.1. *Summarization approach by decomposing tensor T of order 3*

Unfortunately, however, extraction-based methods have already reached a ceiling in terms of their performance, as shown in the HexTAC experiment [GEN 09a] about human extract generation at the TAC'09 campaign. The experiment gave worrying results for system developers: it revealed that people produce better summaries than any systems, even when the latter are limited to extracts. Worse still, when people generate abstracts, they outperform the best human extracts. Extractive methods, the core of the best automatic summarization systems, are insufficient when it comes to achieving the same quality as human summarizers. It is, therefore, necessary to conduct more research into abstraction-based methods. Some systems have taken the plunge using natural language generation modules, information extraction and paraphrasing (via multisentence compression and fusion).

Automatic text summarization is an application discipline in which the final expression is represented by real systems. In spite of the aforementioned difficulties, results are tangible. Indeed, one of the driving forces behind the development of summarization algorithms has been their applications. There are multiple applications for automatic text summarization: news, RSS, email, web page and blog summarization; specialized document, meeting and report summarization; biographic extracts; opinion summarization; automatic title generation etc. Tasks that 50 years ago we would never have dreamed of are today common subjects for automatic summarization systems. The overload of textual information on the Internet will certainly increase in the future, and with it the difficulty (for people) of finding the relevant information they need. Information retrieval is the victim of its own success as classic indexing via keywords is no longer enough; this is a fertile ground for automatic text summarization as it helps human users to save time, provides access to source literature and better targets their research. Automatic text summarization also improves the performance of IR systems.

The scientific community has looked into the problem of evaluating summaries, though it has still not established any definitive solutions. Several partial solutions have been proposed. Evaluations – either intrinsic or extrinsic – concern the form and the content of generated summaries. Intrinsic evaluations (manual, semi-automatic or completely automatic) present both advantages and disadvantages. Evaluations realized by people analyze summaries in detail. However, they are too costly in terms of time and resources. Manually evaluating large corpora of summaries is, therefore, a utopia. Moreover, there is no guarantee that judges will agree. Semi-automatic methods process much larger volumes of summaries to be evaluated; their analysis is not, however, as reliable as that of manual evaluations. These evaluations center principally on the content of summaries and not on questions of form. They also require reference summaries that must be written by people. Finally, automatic evaluations require no human intervention and no references. However, this type of evaluation is for the moment still less reliable than that of the other methods. Research into the automatic evaluation of text summaries is also open.

The automatic summarization systems of the future will be concerned with modalities other than that of text. Notably, multimedia summarization, which generally contains audio documents, images and videos alongside text documents. Currently, systems that try to produce multimedia summaries are based on text summarization techniques. But in addition to the aforementioned problems relating to text, there are particularities of the multimedia dimension. How can a video or an audio extract be evaluated? How can temporality be taken into account? And the redundancy of images or speech?

Fifty years have not been enough to pass the Turing test; neither have they been enough to write a program that is capable of generating abstracts of human quality, nor to develop an elegant text summarization theory. We do not even know if such a theory will ever be created. If we want to produce real abstracts – which are concise, coherent and grammatically correct – in the future, we will probably have to change paradigm. Nevertheless, 50 years have been enough to automatically produce summaries that are very relevant in terms of content. Systems are now capable of processing huge volumes of texts by breaking down the barriers of language, topic and the source of the document. This makes them particularly exploitable by people and machines. Throughout this long journey, advances in research have been measurable and particularly productive. Automatic text summarization has taken its rightful place and has thereby become a precious resource in NLP and IR tasks.

Appendix 1

Information Retrieval, NLP and Automatic Text Summarization

Natural language processing (NLP)[1] and automatic text summarization (ATS) use several techniques from information retrieval (IR) , information extraction (IE) and text mining [BER 04, FEL 07]. Some such techniques are:

– text preprocessing;

– the vector space model (VSM) for the representation of documents;

– similarity measures;

– relevance measures.

In order to understand the issues and algorithms used in NLP and ATS, readers should have prior knowledge of basic IR techniques. For information about IR please consult works by [VAN 79], [BAE 99] and [MAN 08][2] and for information about statistical NLP and IE consult

[1] "Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction" (source: Wikipedia: http://en.wikipedia.org/wiki/Natural_language_processing).

[2] The website http://www-nlp.stanford.edu/IR-book/ is an excellent companion to this book.

Manning and Schütze's book [MAN 99b]. A brief explanation of some of these techniques will be given below.

A.1. Text preprocessing

"In linguistics, a *corpus* (plural corpora) or text corpus is a large and structured set of texts (nowadays usually electronically stored and processed). They are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory."[3] Therefore, text *corpora* are the raw material for NLP software. However, texts often contain misspelt or mistakenly attached words, typographic inconsistencies, ungrammatical sentences, strange characters or they are in a different coding language than that used by the software, etc. Texts are, in a certain sense, "dirty" or noisy. Documents must undergo a suitable "cleaning" and normalization process before they can be processed by NLP algorithms. Therefore, the phase known as text preprocessing is essential. If neglected or realized in a too simplistic manner, systems risk giving biased results. Indexing engines or automatic summarization systems in particular are very sensitive to the amount of noise in a text.

Classically, preprocessing is composed of appropriately splitting the text (into text units such as sentences and words), word normalization and, depending on the application, an adequate filtering of certain terms and punctuations. Preprocessing is a crucial step for NLP systems, which is not evident at all. If interested please consult [PAL 10], which has a very good chapter about text preprocessing.

A.1.1. *Text splitting*

Text documents are generally split into segments. Sentences and words are the most common preferred basic text units. Correctly splitting a text is, in fact, no simple matter. The difficulty of this task is easy to recognize and relevant algorithms must be used to tackle it.

[3] Source: Wikipedia: http://en.wikipedia.org/wiki/Text_corpus.

Sentence splitting

Sentence splitting involves inserting a suitable symbol (for instance </S>) at the end of each recognizable sentence. But the problem is not as simple as it may seem. First, what is a sentence? Where are its boundaries? The following example illustrates the difficulty of answering this question and there being just one splitting choice:

– "You might just as well say", added the March Hare, "that 'I like what I get' is the same thing as 'I get what I like'!"[4]

The three following options for splitting are all possible:

– "You might just as well say",</S> added the March Hare,</S> "that </S> 'I like what I get'</S>, is the same thing as</S> 'I get what I like'!"</S>

– "You might just as well say",</S> added the March Hare,</S> "that 'I like what I get', is the same thing as 'I get what I like'!"</S>

– "You might just as well say", added the March Hare, "that </S> 'I like what I get'</S>, is the same thing as</S> 'I get what I like'!"</S>

A simple approach, which nevertheless works as an elementary system, consists of splitting texts according to full stops, exclamation marks, question marks and colons (. *! ? :*). However, problematic cases such as abbreviations, first names or initials pose a real challenge to splitting algorithms.

Tokenization

"A token is a single instance of a symbol, whereas a type is a general class of tokens" [MAN 08]. In the context of automatic text summarization, it is always useful to correctly count the occurrences of words in texts. This means that systems must be capable of identifying different words (tokens) because they reveal the unique word types in a *corpus*.

For instance, in the sentence: "To have time, to take time... but I won't have the time", the three words "time" correspond to the same

[4] Lewis Carroll, *Alice's Adventures in Wonderland*, Chapter VII.

word type. Therefore, there are three occurrences of the word type “time”.

Consider the following example (from Samuel Beckett):

“Ever tried. Ever failed.
No matter. Try again.
Fail again. Fail better”.

There are two tokens of the word type “ever”, two tokens of the word type “again”, and two tokens of the word type “fail”. However, words which are spelt in the same way but which belong to different grammatical categories are known as homographs. Therefore, in the sentence: “I tie a bow in my hair and bow neatly to the bow of the boat”. the three words “bow” correspond to different word types: a noun or a verb.

Simplistically, an algorithm can split a text into separate tokens using blank spaces, once punctuation has been eliminated. But it is important to pay attention so that the real ends of the sentences and one-character words are respected. Tokenizing a sentence such as: “If I didn’t have time, I wouldn’t go.” can give a list of {if, i, didn, t, have, time, wouldn, go}, where the set of tokens {’t, i} is difficult to manage, particularly because of the lost negation, “n’t”.

A.1.2. *Token normalization*

To reduce the complexity of text representation (such as in a vector model, see Appendix A.2.1), normalization and lexical filtering processes must be realized. The immediate advantage of these processes is that the number of words to be processed is reduced. Generally speaking, the operation which consists of identifying a canonical representation for a set of – similar – words is known as normalization. The concept of normalization is very general. For example, it includes associating words written in different cases and transforming them into lower-case letters or substituting derivations according to their stem. *Grosso modo*, we can speak of morphologically processing stems (lemmatization and stemming) and the normalization of graphic forms. Filtering eliminates terms and stop words[5], as well as punctuation marks and strange characters.

Lemmatization

Lemmatization consists of finding the stem of inflected verbs and putting plural and/or feminine words in the masculine singular form. As such the French terms {*chant, chantaient, chanté, chanteront*} will become "chanter"; {*chanteuse, chanteuses, chanteurs*} will become "chanteur"; and {*chants*} will become "chant". Lemmatization rules can be quite tricky to implement. Algorithms must contain heuristics for regular verb forms (such as {*aime, aimera, aimeront, aimaient*} → "aimer") as well as adequate lists or heuristics for the countless irregularities (for instance {*fut, serons, suis, sera, serait, eusse, été, ...*} → "être"), which are present particularly in romance languages. For example, Flemm[6] is a lemmatizer of French texts which have been parts-of-speech (POS)-tagged (see section A.1.5) by TREETAGGER or Brill tagger.

Stemming

Stemming is the process of eliminating words' suffixes to obtain their common stem. This enables the basic (often truncated) form to be generated. This is known as the stem. For instance: {*computers, computing, computation*} → "comput", {*engineering, engineered, engineer*} → "engineer". One of the most popular algorithms for stemming is [POR 80]'s.[7] Another interesting site is that of the *Lancaster Stemming Algorithm.*[8] *Lemmatization, though more costly in terms of time and resources, has proven to be more suited (than*

[5] "In computing, stop words are words which are filtered out prior to, or after, processing of natural language data (text). There is not one definite list of stop words which all tools use and such a filter is not always used. Some tools specifically avoid removing them to support phrase search" (source: Wikipedia: http://en.wikipedia.org/wiki/Stop_words).

[6] [NAM 00] Flemm is available at https://github.com/tla/cpanmods/tree/master/Flemmv31.

[7] http://tartarus.org/martin/PorterStemmer.

[8] www.comp.lancs.ac.uk/computing/research/stemming/index.htm.

stemming) to Latin languages, such as Spanish, Italian and French, which have a high rate of inflection.

An alternative to lemmatization and stemming is lexematization, that is, clustering words in the same family according to spelling. In principle, lexematization enables the words {*chanter, chantaient, chanté, chanteront, chanteuse, chanteuses, chanteurs, chants*} to be clustered in the same graphic form: “chant”.[9]

Normalization of graphic forms

It is possible that French forms such as “cœur” (heart) or “œufs” (eggs) will not be correctly recognized because of the coding (utf8, ISO-8859, etc.) of the character “œ”. It would be better if they were replaced with “coeur” and “oeufs”, respectively. Accents can also cause difficulties due to the way they have been coded. In IR, accents and tildes are simply deleted. The normalization of cases is also important in this phase. As such, “university”, “UNIVERSITY”, “UnIvErSiTy” and “University” will all be considered forms of “university”. This takes a much too simplistic view of this task. Therefore, readers interested in the disambiguation of graphic forms should consult, for example, [MIK 02].

A.1.3. *Filtering*

The most common words do not, generally speaking, contain a large amount of information. They are called stop words or terms. In IR it is possible for stop words such as articles, conjunctions, figures, punctuation and special symbols[10] to be deleted during filtering. There are typically approximately 200 stop words in English. These terms account for between 20% and 30% of the total word types contained in a typical English text. They can be grouped into a stoplist, which are either domain dependent or independent.[11] For applications in a

[9] A statistical algorithm for lexematization has been developed by [TOR 10].

[10] Such as: &, #, “, ”, [,], |, _, /, {, }, -, =, $, [,], %, @, § etc.

[11] A stoplist is a list words that it is undesirable for a NLP system to process. [VAN 79] and [SAL 71] give examples of domain-independent English stoplists.

specific domain, adequate stoplists composed of stop words can be manually constructed. Table A.1 shows a stoplist of English stopwords from the website http://www.cpan.org [12]. Lists in several languages can be downloaded from http://snowball.tartarus.org/algorithms/.

a	before	down	her	is	on	should	they’d	we’re	won’t
about	being	during	here	isn’t	once	shouldn’t	they’ll	we’ve	would
above	below	each	here’s	it	only	so	they’re	were	wouldn’t
after	between	few	hers	it’s	or	some	they’ve	weren’t	you
again	both	for	herself	its	other	such	this	what	you’d
against	but	from	him	itself	ought	than	those	what’s	you’ll
all	by	further	himself	let’s	our	that	through	when	you’re
am	can’t	had	his	me	ours	that’s	to	when’s	you’ve
an	cannot	hadn’t	how	more	ourselves	the	too	where	your
and	could	has	how’s	most	out	their	under	where’s	yours
any	couldn’t	hasn’t	i	mustn’t	over	theirs	until	which	yourself
are	did	have	i’d	my	own	them	up	while	yourselves
aren’t	didn’t	haven’t	i’ll	myself	same	themselves	very	who	
as	do	having	i’m	no	shan’t	then	was	who’s	
at	does	he	i’ve	nor	she	there	wasn’t	whom	
be	doesn’t	he’d	if	not	she’d	there’s	we	why	
because	doing	he’ll	in	of	she’ll	these	we’d	why’s	
been	don’t	he’s	into	off	she’s	they	we’ll	with	

Table A.1. *Stoplist in English from CPAN* `Lingua::Stopwords`

Statistical filtering, based on a Zipf [13], type criterion, can also be applied to eliminate too-frequent terms. A simple algorithm can make do with generic stoplists to identify terms and symbols to be filtered. Other approaches that use synonyms for reducing lexicon exist, though they present us with a compromise: the introduction of noise.

It must be borne in mind that a filtering lexicon depends on the type of document to be processed. We have discussed filtering generic documents: this has no similarity whatsoever with filtering documents from a specialized domain or documents whose structure is linked to

[12] http://search.cpan.org/creamyg/Lingua-StopWords-0.09/lib/Lingua/StopWords.pm.

[13] “Zipf’s law, an empirical law formulated using mathematical statistics, refers to the fact that many types of data studied in the physical and social sciences can be approximated with a Zipfian distribution, one of a family of related discrete power law probability distributions” (source: Wikipedia: http://en.wikipedia.org/wiki/Zipf’s_law).

the source, such as web pages. In fact, web pages contain non-text elements (images, sound and video), meta-information or hyperlinks, which it is best to eliminate prior to processing. In documents from a specialized domain, such as chemistry, a very specific type of filtering is required. Indeed commas, parentheses, exponents, colons, acronyms and abbreviations, etc. have a different meaning. This is also true for the use of the uppercase (symbols of chemical elements such as He, C, O, etc.). As such the following *corpus*:

> *Prepare this solution by sparging methylene blue in a solution of potassium or sodium tetrachloroplatinate (II),* K_2PtCl_4 + $C_2.H4^+$ = *K[Pt* h^2 *-(*C_2H_4*)*Cl_3*]* H_2O + KCl^-*, and the strange connections of C-Halogen in RI > CBr.NaCl >> Cl are to be taken into account in the reaction: 1/2 Mg(OH)*$_2$ + *NOH where the methyl-2,3-[NaOH]*$_2$ *is sublimated.*

poses enormous challenges to classic text preprocessing systems, as they risk erasing figures, symbols and standardizing the case, which can result in significant information loss. [BOU 08b] presents a combination of statistical and symbolic methods for the recognition of chemical entities: molecule names, elements, commercial names, etc. With regard to documents from the web, please refer to section 6.6.2, which describes the OCELOT system's preprocessing operation for web pages [BER 00]. Please consult these works for a better understanding of the problem of text preprocessing and lexicon filtering for this type of document.

Finally, readers can try systems such as the *Natural Language Toolkit (NLTK)* [BIR 04, LOP 02] *to realize text preprocessing tasks. NLTK contains, for instance, the* PUNKT [KIS 06] algorithm for sentence splitting. NLTK is composed of free modules in Python for NLP and can be downloaded from http://www.nltk.org/. Steven Bird, Ewan Klein and Edward Loper's book, "Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit", *O'Reilly Media* 2009, provides a good introduction to NLTK and is available at the above address.

A.1.4. *Named entities*

Locating named entities (NE) is a process that consists of automatically detecting the names of people, organizations, places (cities, countries, etc.) and physical and temporal quantities. This task enables semantic tags to be added to the text. However, the border between what is (and is not) a named entity is still disputed and its definition remains opaque. In order to clarify this situation, the Message Understanding Conference (MUC) defined a hierarchy of NEs which in principle facilitates their extraction:

– ENAMEX: proper nouns,

– TIMEX: temporal expressions,

– NUMEX: numerical expressions and percentages,

which is in turn split into subhierarchies.

A system for detecting NEs in English is *The Stanford Natural Language Processing Group*: the Stanford Named Entity Recognizer (NER). It is based on Conditional Random Fields classifiers (CRFClassifier) and is downloadable from: http://nlp.stanford.edu/software/CRF-NER.shtml. A system for recognizing NEs in French is CasEN,[14] which is based on finite-state transducer cascades. Finally, an online commercial system for NE detection is Wikimeta, available at http://www.wikimeta.fr, and available in several languages.

A.1.5. *POS-tagging*

Morpho-syntactic tagging, also known as grammatical tagging or POS-tagging, is a process that consists of automatically associating words in a text with corresponding grammatical information: verbs, adjectives, nouns, gender, number, etc. Several models have been developed to grammatically annotate texts: Hidden Markov Model (HMM), entropy, perceptrons, decision trees, etc. A free and very

[14] *Cascade de transducteur CasEN pour la reconnaissance des entités nommées* is available at: http://tln.li.univ-tours.fr/Tln_CasEN.html; [FRI 04].

popular system for realizing POS-tagging and lemmatization is TreeTagger by [SCH 94]. The Association for Computational Linguistics (ACL) wiki page[15] describes a relatively complete state-of-the-art with resources, articles, hyperlinks, etc. concerning POS-tagging.

For instance, for the sentence:

"A man can be happy with any woman as long as he does not love her."[16]

a possible POS-tagging is:

A/DT man/NN can/MD be/VB happy/JJ with/IN any/DT woman/NN as/RB long/RB as/IN he/PP does/VBZ not/RB love/VB her/PP ./SENT

where the tag /NN represents a noun, /VB a verb, JJ/ an adjective and so on and so forth.

Ambiguous sentences of course cause systems great difficulties as the same word can be associated with tags of different categories. Therefore, a significant percentage of grammatical tags can be ambiguous. For instance, for the classic sentence:

"The Duchess was entertaining last night."[17]

the word *entertaining* can function either as an adjective or as a verb. POS-tagging carried out by the TreeTagger system is:

```
The           DT    the
Duchess       NP    <unknown>
was           VBD   be
```

[15] http://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art).

[16] Oscar Wilde.

[17] Beatrice Santorini, "Part-of-speech Tagging Guidelines for the Penn Treebank Project", (3rd rev, June 1990), http://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis_reports.

```
entertaining  VBG   entertain
last          JJ    last
night         NN    night
.             SENT  .
```

A.2. The vector space model

A.2.1. *A matrix representation*

The vector space model (VSM) [SAL 68, SAL 83, SAL 89] has been widely used in IR, document classification, information filtering and automatic text summarization. This model seeks to find the relation between the N documents in a collection and a set of k characteristics in this collection. It is also known as the bag-of-words model, as the representation pays no attention to the order of the words in a document and, is instead, only concerned with their relations. The vector representation of texts (or vectorization of texts), although very different to structural linguistic analysis, has proven to be a high-performing model [MAN 99b]. Vector models also have the property of being quite independent from language where texts are processed as a bag of terms. Terms are words or n-grams. An n-gram is a subsequence of n elements (words, for example), constructed from a given sequence (a sentence, for example). Sequences that are two words long are known as bigrams. Sequences that are three words long are trigrams and so on and so forth. n-grams can be constructed via a “sliding window” of words of a length n that covers the sentence. The number of combinations of n-grams is higher than the number of words in a sentence.

In IR, the vectorization of documents creates a document-term matrix where each column represents the weight of a term in a sentence (or a document). The matrices that result from the vectorization of documents are very voluminous, sparsed and extremely noisy. This means that the relevant information is only a small part of the total volume of available data [LEB 04]. This is why filtering and normalization mechanisms are indispensable for reducing the complexity of the lexicon and thus the dimension of this matrix [MAN 99b, MAN 08].

DEFINITION APPENDIX 1.1.– VECTOR MODEL. *Let* $\{t_1, t_2, \ldots, t_k\}$ *be the set of terms and* $\{D_1, D_2, \ldots, D_N\}$ *the set of documents. A document* D_i *is represented as a vector:*

$$\vec{D_i} = \{w(D_i, t_1), w(D_i, t_2), \ldots, w(D_i, t_k)\} \qquad [A.1]$$

where $w(D_i, t_j)$ *is the weight of a term* t_j *in the document* D_i*.*

There are several ways of weighting terms in the vector model. One of the most popular ways is the product of tf.idf (*Term Frequency, Inverse Document Frequency*) [SPÄ 72]. The product of tf.idf is commonly used in IR.

DEFINITION APPENDIX 1.2.– *The tf.idf of a term* j *in a corpus is defined by:*

$$\textit{tf.idf}_{i,j} = \textit{tf}_{i,j} \cdot \log_2 \frac{N}{n} \qquad [A.2]$$

where $\textit{tf}_{i,j}$ *is the frequency of the term* j *in a document* i*;* N *is the total number of documents in the corpus and* n *is the number of documents in which the term* j *appears.*

The normalized version of tf.idf is:

$$w(D_i, t_j) = w_{i,j} = \frac{\mathrm{tf}_{i,j} \cdot \mathrm{idf}_j}{\parallel \vec{D_i} \parallel} = \frac{\mathrm{tf}_{i,j} \cdot \log_2 \frac{N}{n_j}}{\sqrt{\sum_{s=1}^{k} (\mathrm{tf}_{i,s} \cdot \log_2 \frac{N}{n_j})^2}} \qquad [A.3]$$

where $\mathrm{tf}_{i,j}$ is the number of times the term t_j appears in D_i and n_j is the number of documents in which t_j appears. If a term j is frequent in a document i, it is assumed to be important and its $\mathrm{tf}_{i,j}$ increases. Even if this term appears in several documents (n) it is not discriminatory and its inverse frequency in the idf documents decreases:

$$\mathrm{idf}_j = \log_2 \frac{N}{n_j} \qquad [A.4]$$

Similarly, another weighting method is inverse term frequency itf_i for a document i, defined below:

$$\text{itf}_i = \log_2 \frac{k}{n'} \qquad \text{[A.5]}$$

where n' is the number of terms in which the document D_i “appears”.

The classic vector model is very popular as it is both simple and easy to implement. It also gives good results. This model must be preceded by text preprocessing (described in section 2.1 and Appendix A.1) to improve its performance. There are other models for text representation, such as the binary model, the extended vector model and probability models; these models are less used. If interested, [BAE 99] contains an in-depth discussion of these models.

An example of calculating tf.idf and tf.itf weights will be presented below. Let us suppose that the document-term matrix of Table A.2 was generated from the vectorization of a *corpus* of $N = 4$ documents. Tables A.3 and A.4 show the calculation of inverse occurrences n and the calculation of tf.idf of each term respectively.

	mouse	cat	chips	cheese	computer	linux
D_1	1	1	0	1	0	0
D_2	1	1	0	0	1	1
D_3	2	0	1	0	2	1
D_4	1	1	1	1	0	0

Table A.2. *Document-term matrix*

Term j	mouse	cat	chips	cheese	computer	linux
Occurrences tf_j	5	3	2	2	3	2
Docs containing term j	4	3	2	2	2	2

Table A.3. *Computing df on* $k = 6$ *terms*

Table A.5 shows the inverse occurrences n' and the calculation of df.itf by document.

j	Term	Occurrences tf_j	n_j	Inverse Document Frequency $\text{idf}_j = \log_2(N/n_j)$	$w_j = \text{tf}_j.\text{idf}_j$
1	mouse	5	4	$\log_2(4/4) = \log_2 1.00 = 0.000$	$5 \times 0.000 = 0.00$
2	cat	3	3	$\log_2(4/3) = \log_2 1.33 = 0.415$	$3 \times 0.415 = 1.25$
3	chips	2	2	$\log_2(4/2) = \log_2 2.00 = 1.000$	$2 \times 1.000 = 2.00$
4	cheese	2	2	$\log_2(4/2) = \log_2 2.00 = 1.000$	$2 \times 1.000 = 2.00$
5	computer	3	2	$\log_2(4/2) = \log_2 2.00 = 1.000$	$3 \times 1.000 = 3.00$
6	linux	2	2	$\log_2(4/2) = \log_2 2.00 = 1.000$	$2 \times 1.000 = 2.00$

Table A.4. *Calculation of tf.idf of $k = 6$ terms, $N = 4$*

i	Document	Occurrences df_i	n'_i	Inverse Term Frequency $\text{itf}_i = \log_2(k/n'_i)$	$w_i = \text{df}_i.\text{itf}_i$
1	D_1	3	3	$\log_2(6/3) = \log_2 2.0 = 1.00$	$3 \times 1.00 = 3.00$
2	D_2	4	4	$\log_2(6/4) = \log_2 1.5 = 0.58$	$4 \times 0.58 = 2.34$
3	D_3	6	4	$\log_2(6/4) = \log_2 1.5 = 0.58$	$6 \times 0.58 = 3.51$
4	D_4	4	4	$\log_2(6/4) = \log_2 1.5 = 0.58$	$4 \times 0.58 = 2.34$

Table A.5. *Calculation of df.itf of $N = 4$ documents, $k = 6$*

A.2.2. *The similarity of documents*

There are several ways of calculating the similarity of documents. Four very popular measures will be presented: cosine similarity (and tf.idf-based cosine similarity), Euclidean distance, Jaccard similarity and Dice coefficient.

Cosine similarity

The similarity between two documents can be calculated using a scalar product or cosine similarity:

$$sim(\vec{D_1}, \vec{D_2}) = \langle \vec{D_1} \ldots \vec{D_2} \rangle = \sum_{j=1}^{k} w_{1,j} \times w_{2,j} \qquad \text{[A.6]}$$

where $\vec{D_1}$ and $\vec{D_2}$ are the vectors of documents 1 and 2, respectively, and $w_{\bullet,j}$ is the weight of term j in the document $\bullet$. Geometrically, the scalar product corresponds to the cosine of the angle θ between vectors $\vec{D_1}$ and $\vec{D_2}$, divided by their norm:

$$\cos\theta = \cos(\vec{D_1}, \vec{D_2}) = \frac{\langle \vec{D_1} \ldots \vec{D_2} \rangle}{\parallel \vec{D_1} \parallel \times \parallel \vec{D_2} \parallel} \qquad \text{[A.7]}$$

where $\parallel \ \vec{x} \ \parallel = \sqrt{x_1^2 + x_2^2 + \ldots + x_k^2}$. Cosine is a very popular similarity measure in IR. Cosine similarity is always a value between [0,1], which enables it to be considered as having already been normalized. It is important to point out that two identical documents have a similarity of 1. Conversely, they are orthogonal if they do not contain the same terms and, thus, their similarity is zero.

According to the example in Table A.2, the cosine similarity between D_1 and the other documents is: $\cos(D_1, D_2) = \frac{2}{\sqrt{3}\times\sqrt{4}} = 0.577$; $\cos(D_1, D_3) = \frac{2}{\sqrt{3}\times\sqrt{10}} = 0.365$; and $\cos(D_1, D_4) = \frac{3}{\sqrt{3}\times\sqrt{4}} = 0.866$. Thus, D_1 is close to D_4.

Idf-modified-cosine similarity between two sentences is calculated by [ERK 04]:

$$\text{idf-modified-cos}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{x,w}\text{tf}_{y,w}(\text{idf}_w)^2}{\sqrt{\sum_{x_j \in x}(\text{tf}_{x,x_j}\text{idf}_{x_j})^2} \times \sqrt{\sum_{y_j \in y}(\text{tf}_{y,y_j}\text{idf}_{y_j})^2}} \qquad \text{[A.8]}$$

where $\text{tf}_{s,w}$ is the number of occurrences of the word w in the sentence s.

Euclidean distance

The Euclidean distance between two documents is given by formula:

$$dist_E(D_1, D_2) = ||D_1 - D_2|| = \sqrt{\sum_{j=1}^{k}(w_{1,j} - w_{2,j})^2} \qquad \text{[A.9]}$$

Figure A.1 shows a document R (a query or a reference summary) and two documents D_1 and D_2, in a dimensional space of $k = 3$ terms. In cosine similarity, the document D_2 is closer to R because its angle θ is less than the angle α between the document D_1 and the document R. The Euclidean distance between D_1 and R will also be shown.

In Table A.2, the euclidean similarity between D_1 and the other documents is:

– $dist_E(D_1, D_2) = \sqrt{0+0+0+1+1+1} = \sqrt{3}$;

– $dist_E(D_1, D_3) = \sqrt{9}$;

– $dist_E(D_1, D_4) = \sqrt{1}$.

Thus, D_1 is more similar to D_4.

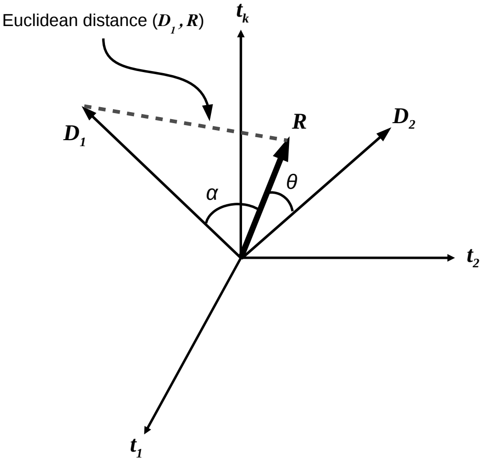


Figure A.1. *Similarity between documents D_1, D_2 and a query R*

Jaccard similarity

In statistics, the Jaccard index [18] enables the similarity between two sets – A and B – to be evaluated. The Jaccard index is the relation

[18] "The Jaccard index, also known as the Jaccard similarity coefficient (originally coined coefficient de communauté by Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets" (source Wikipedia: http://en.wikipedia.org/wiki/Jaccard_index).

between the cardinality of the intersection of A and B and the cardinality of the union of these sets:

$$i_J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad \text{[A.10]}$$

In IR, the Jaccard index of similarity between two documents D_1, D_2 corresponds to the number of common words divided by the total number of words minus the number of common words:

$$i_J(D_1,D_2) = \frac{m_c}{m_1 + m_2 - m_c} \qquad \text{[A.11]}$$

where m_c is the number of common words or shared lexicon; m_1 is the size of the lexicon in the document D_1 (the number of word types in D_1); m_2 is the size of the lexicon in the document D_2; m_1 + m_2 is the sum of the lexicons in D_1 and D_2.

In Table A.2, the similarity between D_1 and the other documents is:

$i_J(D_1,D_2) = \frac{2}{3+4-2} = 0.40$; $i_J(D_1,D_3) = \frac{1}{3+4-1} = 0.17$; $i_J(D_1,D_4) = \frac{3}{3+4-3} = 0.75$. D_1 is thus more similar to D_4.

Dice coefficient

The Dice coefficient is the relation between the cardinality of the intersection of A and B and the cardinality of the union of these sets:

$$i_D(A,B) = \frac{2 \times |A \cap B|}{|A| + |B|} \qquad \text{[A.12]}$$

In IR, the Dice coefficient measures the similarity between two documents D_1 and D_2 based on the number of common words:

$$i_D(D_1,D_2) = \frac{2 \times m_c}{m_1 + m_2} \qquad \text{[A.13]}$$

where m_c is the number of common words between D_1 and D_2, m_1 is the number of words in D_1 and m_2 is the number of words in D_2.

For example, in Table A.2, the similarity between D_1 and the other documents is: $i_D(D_1, D_2) = \frac{2\times 2}{3+4} = 0.571$; $i_D(D_1, D_3) = \frac{2\times 1}{3+4} = 0.286$; and $i_D(D_1, D_4) = \frac{2\times 3}{3+4} = 0.857$. D_1 is thus more similar to D_4.

The Jaccard and Dice similarities are 0 when word overlap is null and 1 when there is perfect overlap. The Jaccard index penalizes sentences with few common words more than the Dice index.

A.3. Precision, recall, F-measure and accuracy

Historically, relevance measures in IR are precision and recall. A *receiver operating characteristic* (ROC) curve [19] of precision *versus* recall shows the performance of a system. These two measures can be combined in an arithmetic mean known as the F-measure (or F-score). Another commonly used measure is the accuracy of a system.

In the context of IR, the precision P, the recall R and the F-measure are defined in terms of a set of retrieved documents (for example, the list of documents produced by an online search engine following a query) and a set of relevant documents (the list of documents retrieved from the internet which are relevant to a given subject). P and R can be harmonically combined via a modulable parameter β, in the F-measure.

DEFINITION APPENDIX 1.3.– *Precision P is the number of relevant documents in the retrieved documents:*

$$P = \frac{|\{\textit{relevant documents}\} \cap \{\textit{retrieved documents}\}|}{|\{\textit{retrieved documents}\}|} \qquad [A.14]$$

DEFINITION APPENDIX 1.4.– *Recall R is the number of retrieved documents in the relevant documents:*

$$R = \frac{|\{\textit{relevant documents}\} \cap \{\textit{retrieved documents}\}|}{|\{\textit{relevant documents}\}|} \qquad [A.15]$$

[19] Source: http://en.wikipedia.org/wiki/Receiver_operating_characteristic.

DEFINITION APPENDIX 1.5.– *F-measure is a harmonic mean combining recall and precision. The value β is set according to the target application:*

$$F\text{-}measure(\beta) = \frac{(1+\beta^2)\cdot P \cdot R}{\beta^2 \cdot P + R} \qquad [A.16]$$

If $\beta = 1$ neither recall nor precision is favored:

$$F\text{-measure}(\beta = 1) = \frac{2 \cdot P \cdot R}{P + R} \qquad [A.17]$$

A system is considered perfect if $P = R = 1$. However, if all the documents are retrieved then $R = 1$ and $P \approx 0$. If no document is retrieved, $P = 1$ and $R \approx 0$. It is common to use different measure thresholds (5, 10, 20,... first documents returned) for the graphic representation of precision and recall.

Another representation of performance is the confusion matrix. A confusion matrix is used to measure the quality of a classification system. “In the field of machine learning, a confusion matrix, also known as a contingency table or an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix).”[20] The four cells in the confusion matrix (see the matrix in Table A.6) represent the number of occurrences of true positives *(TP)* (positive instances • that have been correctly classified by the system), false positives *(FP)* (negative instances ○ that have been classified as positive by the system), true negatives *(TN)* (negative instances ○ that have been correctly classified as negative by the system) and false negatives *(FN)* (positive instances • that have been classified as negative by the system).

[20] Source: Wikipedia http://en.wikipedia.org/wiki/Confusion_matrix.

	Actual class ●	Actual class ○
Predicted class ●	TP	FP
Predicted class ○	FN	TN

Table A.6. *Confusion matrix*

In the confusion matrix, the $TP{+}FP = 100\,\%$ and the $FN{+}TN = 100\%$ of instances. Precision P, recall R and accuracy ACC, can be derived from this matrix:

$$P = \frac{TP}{TP+FP} \quad ; \quad R = \frac{TP}{TP+FN} \qquad \text{[A.18]}$$

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \qquad \text{[A.19]}$$

Below is an example of calculating precision, recall and F-measure in terms of IR. Let two summarization algorithms be applied to the task of summarization by extraction. The hypothetical results of the output of the algorithms on a document composed of $\rho = 10$ sentences, as well as the extraction generated by a human judge are shown in Table A.7.

Sentence	Human judge	Summarizer$_1$	Summarizer$_2$
1	●	●	●
2	○	●	○
3	●	●	●
4	●	○	○
5	○	○	●
6	○	○	○
7	○	○	○
8	○	○	○
9	○	○	○
$\rho = 10$	●	●	○

Table A.7. *Output of two summarizers ● extracted sentence, ○ non-extracted ($\neg$) sentence*

The most interesting question is: which of the two summarizers is the higher performing? (evidently, performance is measured by the task of correctly extracting the sentences retained by the judge). In this table, the total number of instances is $\rho = 10$, the number of instances in the "extracted" class is $n(\bullet) = 4$ and the number of instances in the "non-extracted" class is $n(\circ) = 6$.

An analysis of the summarizer$_1$ gives:

– out of the four "extracted" sentences, three are correctly extracted and one is judged "non-extracted": $TP = 3$ and $FP = 1$;

– out of the six "non-extracted" sentences, one is incorrectly considered and five are judged "non-extracted": $TN = 5$ and $FN = 1$.

For summarizer$_2$, out of three "extracted" sentences, two are deemed as such and one is judged "non-extracted": $TP = 2$ and $FP = 1$. Out of the seven "non-extracted" sentences, two are considered to be "extracted" and five are judged as "non-extracted": $TN = 5$ and $FN = 2$. This enables the confusion matrix for each summarizer (see Table A.8) to be generated.

Summarizer$_1$	Human judge		Summarizer$_2$	Human judge	
	Extr. ●	¬ Extr. ○		Extr. ●	¬ Extr. ○
Extracted ●	3	1	Extracted ●	2	1
¬ Extracted ○	1	5	¬ Extracted ○	2	5

Table A.8. *Confusion matrix of summarizer$_1$ and summarizer$_2$*

From the corresponding confusion matrix, the precision P, recall R and F-measure of the "extracted" class ($\bullet$) of summarizer$_1$ are calculated as follows:

$$P_1(\bullet) = \frac{TP}{TP + FP} = \frac{3}{3+1} = 0.750\,; R_1(\bullet) = \frac{TP}{TP + FN}$$
$$= \frac{3}{3+1} = 0.750$$

$$F\text{-measure}_1(\bullet) = 2 \cdot \frac{P_1 \times R_1}{P_1 + R_1} = 2 \cdot \frac{0.75 \times 0.75}{0.75 + 0.75} = 0.750$$

The accuracy of summarizer$_1$ is:

$$ACC_1 = \frac{TP + TN}{TP + FP + TN + FN} = \frac{3 + 5}{3 + 1 + 5 + 1} = \frac{8}{10} = 0.800$$

The precision P and recall R of the "extracted" class of summarizer$_2$ is:

$$P_2(\bullet) = \frac{TP}{TP + FP} = \frac{2}{2 + 1} = 0.666\,; R_2(\bullet) = \frac{TP}{TP + FN}$$
$$= \frac{2}{2 + 2} = 0.500$$

its F-measure$_2(\bullet) = 0.571$ and its accuracy ACC_2 is:

$$ACC_2 = \frac{2 + 5}{2 + 1 + 2 + 5} = \frac{7}{10} = 0.700$$

From these calculations it can be deduced that summarizer$_1$ is slightly better performing in terms of F-measure than summarizer$_2$, when the "extracted" class is evaluated. Finally, with regard to the P, R and F-measure of the "non-extracted" class ($\circ$), a symmetrical calculation gives:

– summarizer$_1$: $P_1(\circ) = 0.833$; $R_1(\circ) = 0.833$; F-measure$_1(\circ) = 0.833$;

– summarizer$_2$: $P_2(\circ) = 0.714$; $R_2(\circ) = 0.833$; F-measure$_2(\circ) = 0.769$.

This confirms the fact that summarizer$_1$ is better performing than summarizer$_2$ when the "non-extracted" sentences are evaluated. All confused sentences, the mean $\langle F$-measure$\rangle$ of a summarizer is calculated as the fraction of the sum of the i micro-averages, $i = 1, 2, \ldots, N_c$ over the number of classes N_c:

$$\langle F\text{-measure}\rangle = \frac{\sum_i F\text{-measure}_i}{N_c} \qquad [A.20]$$

Which gives, in this example where $N_c = 2$:

– summarizer$_1$: $\langle F$-measure$\rangle_1 = (0.750 + 0.833)/2 = 0.791$;

– summarizer$_2$: $\langle F\text{-measure}\rangle_2 = (0.571 + 0.769)/2 = 0.670$.

Therefore, the mean$\langle F\text{-measure}\rangle$ confirms that summarizer$_1$ is better performing than summarizer$_2$.

In automatic text summarization, the semi-automatic evaluation ROUGE[21] uses the calculation of recall – rather than precision – to count the intersection of co-occurrences of the n-grams contained in the optimal summaries (references written by people), in relation to candidate summaries (created by systems). The higher the recall, the closer (in terms of information content) the candidate summary is to the reference summaries. The automatic evaluation approaches SIMETRIX and FRESA[22] also use recall to measure the divergence of the probability distributions of the n-grams present in the candidate summary in relation to the source: the smaller the divergence, the closer the summary is to the source. Please refer to Chapter 8 for more information about summarization evaluation systems.

[21] Recall-Oriented Understudy for Gisting Evaluation.

[22] Framework for Evaluating Summaries Automatically.

Appendix 2

Automatic Text Summarization Resources

This appendix contains a list of the automatic summarization systems, summarization evaluation systems and *corpora* available on the web. The most representative conferences, congresses, seminars and workshops of the domain are also listed.

Website for the book: http://ats.talne.eu

Online automatic text summarization systems:

– AUTOSUMMARIZER: http://www.autosummarizer.com

– CONNEXOR: http://www.connexor.com

– COPERNIC SUMMARIZER: http://www.copernic.com/en/products/summarizer

– CORTEX: http://daniel.iut.univ-metz.fr/wiso/cortex.php

– DUTCH SUMMARIZER: http://www.clips.uantwerpen.be/sumdemo

– ESSENTIAL SUMMARIZER: http://essential-mining.com/

– EXTRACTOR: http://www.extractor.com

– FOCISUM: http://www1.cs.columbia.edu/hjing/sumDemo/FociSum

– FREE SUMMARIZER: http://freesummarizer.com/

– INTEXT: (Dynamic Summarizing): http://www.intext.com

– INTELLEXER: http://summarizer.intellexer.com/index.html

– ISLANDINTEXT: http://www.islandsoft.com/products.html

– LEXRANK: http://clair.si.umich.edu/clair/demos/lexrank

– MEAD: http://www.summarization.com/mead/

– OPEN TEXT SUMMARIZER (OTS): http://libots.sourceforge.net

– SMMRY SUMMARIZER: http://smmry.com/

– SUMMA Toolkit: http://www.taln.upf.edu/pages/summa.upf/

– SWESUM SUMMARIZER: http://swesum.nada.kth.se/index-eng.html

– SYSTEM QUIRK: http://www.mcs.surrey.ac.uk/SystemQ

– TAKAHE: https://github.com/boudinfl/takahe

– TEXTANALYST: http://www.megaputer.com

– TEXT COMPACTOR: http://textcompactor.com

– TOOLS4NOOBS: http://www.tools4noobs.com/summarize/

– TRESTLE: http://nlp.shef.ac.uk/trestle

– YACHS2: http://daniel.iut.univ-metz.fr/yachs

Text summarization evaluation systems

– BASIC ELEMENTS: http://www.isi.edu/publications/licensed-sw/BE/ index.html

– FRESA: http://fresa.talne.eu/

– MEAD EVAL: http://www.summarization.com/mead/

– PYRAMID: http://www.cs.columbia.edu/ani/DUC2005/

– ROUGE: http://www.berouge.com/Pages/default.aspx

– SIMETRIX: http://homepages.inf.ed.ac.uk/alouis/IEval2.html

Corpus

– DMOZ (web page/gist): http://www.dmoz.org/

– DUC: http://www-nlpir.nist.gov/projects/duc/data.html

– TAC: http://www.nist.gov/tac/data/index.html

– HexTAC: http://rali.iro.umontreal.ca/rali/?q=fr/hextac

– Multiling: http://multiling.iit.demokritos.gr

– RPM2: http://lia.univ-avignon.fr/fileadmin/documents/rpm2/rpm2_resumes_fr.html

– RPM2 sentence compression:
http://lia.univ-avignon.fr/fileadmin/documents/rpm2/rpm2_compresse.html

– Sentence fusion: https://github.com/boudinfl/lina-msc

– Text Summarization, Dragomir Radev's website: http://www.summarization.com/

Conferences and workshops

– Association for Computational Linguistics (ACL); http://portal.aclweb.org

– ACL'14: The 52nd Meeting of the Association for Computational Linguistics; http://www.cs.jhu.edu/ACL2014

– CICLING Conference on Intelligent Text Processing and Computational Linguistics (since 2000); http://www.cicling.org

– CLT'14: 5th Conference on Language and Technology; http://cs.dsu.edu.pk/clt14

– COLING'14: 25th International Conference on Computational Linguistics; http://www.coling-2014.org

– CONLL'14 18th Conference on Computational Natural Language Learning; http://www.conll.org

– EMNLP'14: Conference on Empirical Methods in Natural Language Processing; http://emnlp2014.org

– INEX Tweet Contextualisation Task; https://inex.mmci.uni-saarland.de/tracks/qa/

– KONVENS'14: 12th Konferenz zur Verarbeitung Natürlicher Sprache/Conference on Natural Language Processing; http://www.uni-hildesheim.de/konvens2014

– LTCSS'14: ACL Workshop on Language Technology and Computational Social Science; http://www.mpi-sws.org/cristian/LACSS_2014.html

– NIST Document Understanding Conferences (from 2001–2007); http://duc.nist.gov

– NIST Text Analysis Conferences (since 2008); http://www.nist.gov/tac

– NIST TREC (Text REtrieval Conferences); http://trec.nist.gov

– NTCIR Text Summarization Challenge 2 (TSC, Japan); http://research.nii.ac.jp/ntcir/ntcir-ws3/tsc/cfp-en.html

– TALN-ATALA: Traitement Automatique du Langage Naturel/Automatic Natural Language Processing (since 1994); http://www.atala.org/-Conference-TALN-RECITAL-

– TextGraphs-8 Graph-based Methods for NLP; http://www.textgraphs.org

– Tipster Summac/Text Summarization Evaluation Conference (end 1998); http://www-nlpir.nist.gov/related_projects/tipster_summac/index.html

– TSD'14: Text, Speech and Dialogue; http://www.tsdconference.org/tsd2014

– Wiki for Call For Papers http://www.wikicfp.com/cfp/call?conference=NLP&page=1

Bibliography

[AFA 05] AFANTENOS S., KARKALETSIS V., STAMATOPOULOS P., "Summarization of medical documents: a survey", *Artificial Intelligence in Medicine*, vol. 33, no. 2, pp. 157–177, 2005.

[AFA 10] AFANTENOS S., DENIS P., MULLER P., *et al.*, "Learning recursive segments for discourse parsing", *LREC '10*, ELRA, Malta, pp. 3578–3584, 2010.

[ALO 03] ALONSO L., FUENTES M., "Integrating cohesion and coherence for automatic summarization", *European Chapter of the ACL (EACL '03)*, ACL, Budapest, Hungary, pp. 1–8, 2003.

[ALT 90] ALTERMAN R., BOOKMAN L.A., "Some computational experiments in summarization", *Discourse Processes*, vol. 13, no. 2, pp. 143–174, 1990.

[ANS 79] ANSI, American National Standard for Writing Abstracts Z39-14, ANSI, New York, 1979.

[AON 99] AONE C., GORLINSKY J., LARSEN B., *et al.*, "A trainable summarizer with knowledge acquired from robust NLP techniques", *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pp. 71–80, 1999.

[ARC 13] ARCHANA A., SUNITHA C., "An overview on document summarization techniques", *International Journal on Advanced Computer Theory and Engineering*, vol. 1, no. 2, pp. 113–118, 2013.

[ARE 94] ARETOULAKI M., "Towards a hybrid abstract generation system", *International Conference on New Methods in Language Processing*, Manchester, UK, pp. 220–227, 1994.

[ARE 96] ARETOULAKI M., COSY-MATS: a hybrid connectionist-symbolic approach to the pragmatic analysis of texts for their automatic summarization, PhD Thesis, University of Manchester, Institute of Science and Technology, Manchester, UK, 1996.

[BAE 99] BAEZA-YATES R., RIBEIRO-NETO B., *Modern Information Retrieval*, Addison-Wesley Longman, Boston, MA, 1999.

[BAN 00] BANKO M., MITTAL V.O., WITBROCK M.J., "Headline generation based on statistical translation", *38th Meeting on Association for Computational Linguistics (ACL '00)*, ACL, Hong Kong, China, pp. 318–325, 1–8 October 2000.

[BAN 10] BANEA C., MOSCHITTI A., SOMASUNDARAN S., *et al.*, "TextGraphs-5: graph-based methods for natural language processing", *48th Meeting of the Association for Computational Linguistics (ACL '10)*, ACL, Uppsala, Sweden, 11–16 July 2010.

[BAR 97] BARZILAY R., ELHADAD M., "Using lexical chains for text summarization", *Workshop on Intelligent Scalable Text Summarization (ACL '97/EACL '97)*, Madrid, Spain, pp. 10–17, 11 July 1997.

[BAR 99] BARZILAY R., MCKEOWN K.R., ELHADAD M., "Information fusion in the context of multi-document summarization", *37th Meeting of the Association for Computational Linguistics (ACL '99)*, College Park, MA, pp. 550–557, 20–26 June 1999.

[BAR 02] BARZILAY R., ELHADAD N., MCKEOWN K.R., "Inferring strategies for sentence ordering in multidocument news summarization", *Journal of Artificial Intelligence Research*, vol. 17, no. 1, pp. 35–55, August 2002.

[BAR 05] BARZILAY R., MCKEOWN K.R., "Sentence fusion for multidocument news summarization", *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, 2005.

[BAX 58] BAXENDALE P.B., "Machine-made index for technical literature: an experiment", *IBM Journal of Research Development*, vol. 2, no. 4, pp. 354–361, October 1958.

[BER 00] BERGER A.L., MITTAL V.O., "OCELOT: a system for summarizing web pages", *23rd International Conference on Research and Development in Information Retrieval (SIGIR '00)*, Athens, Greece, ACM Press, New York, pp. 144–151, 2000.

[BER 04] BERRY M.W., *Survey of Text Mining: Clustering, Classification, and Retrieval*, Springer-Verlag, New York, 2004.

[BIR 04] BIRD S., LOPER E., "NLTK: the natural language toolkit", *42nd Meeting of the Association for Computational Linguistics (ACL '04)*, ACL, Barcelona, Spain, pp. 214–217, 21–26 July 2004.

[BLA 03] BLATZ J., FITZGERALD E., FOSTER G., *et al.*, Confidence estimation for machine translation, report, Johns Hopkins University, Baltimore, MD, 2003.

[BLA 04] BLAIR-GOLDENSOHN S., EVANS D., HATZIVASSILOGLOU V., *et al.*, "Columbia University at DUC 2004", *Document Understanding Conference (DUC '04)*, Boston, MA, 2004.

[BOG 97] BOGURAEV B., KENNEDY C., "Salience-based content characterisation of text documents", *Workshop on Intelligent Scalable Text Summarization (ACL '97/EACL '97)*, Madrid, Spain, pp. 2–9, 11 July 1997.

[BOR 75] BORKO H., BERNIER C., *Abstracting Concepts and Methods*, Academic Press, New York, 1975.

[BOS 09] BOSSARD A., GÉNÉREUX M., POIBEAU T., "CBSEAS, a summarization system – integration of opinion mining techniques to summarize blogs", LASCARIDES A., GARDENT C., NIVRE J. (eds.), *The Association for Computer Linguistics*, EACL (Demos), pp. 5–8, 2009.

[BOU 07a] BOUDIN F., TORRES-MORENO J.-M., "NEO-CORTEX: a performant user-oriented multi-document summarization system", *International Conference Computational Linguistics and Intelligent Text Processing (CICLing '07)*, Mexico, Springer-Verlag, Berlin, pp. 551–562, 2007.

[BOU 07b] BOUDIN F., BECHET F., EL-BÈZE M., *et al.*,"The LIA summarization system at DUC-2007", *Document Understanding Conference (DUC '07)*, NIST, Rochester, NY, 2007.

[BOU 08a] BOUDIN F., Exploration d'approches statistiques pour le résumé automatique de texte, PhD Thesis, Laboratoire Informatique d'Avignon, Université d'Avignon et des Pays de Vaucluse, Avignon, France, 2008.

[BOU 08b] BOUDIN F., TORRES-MORENO J.-M., EL-BÈZE M., "Mixing statistical and symbolic approaches for chemical names recognition", *9th International Conference Computational Linguistics and Intelligent Text Processing (CICLing '08)*, Haifa, Israel, Springer-Verlag, Berlin, pp. 334–343, 17–23 February 2008.

[BOU 08c] BOUDIN F., TORRES-MORENO J.-M., VELAZQUEZ-MORALES P., "An efficient statistical approach for automatic organic chemistry summarization", *6th International Conference on Natural Language Processing (GoTAL '08)*, LNAI, Gothenburg, Sweden, vol. 5221, pp. 89–99, 2008.

[BOU 09a] BOUDIN F., TORRES-MORENO J.-M., "A maximization-minimization approach for update summarization", NICOLOV N., ANGELOVA G., MITKOV R. (eds.), *Recent Advances in Natural Language Processing V. Current Issues in Linguistic Theory 309*, vol. 309, RANLP, John Benjamins, Amsterdam, pp. 143–154, 2009.

[BOU 09b] BOUDIN F., TORRES-MORENO J.-M., "Résumé automatique multi-document et indépendance de la langue: une première évaluation en français", *Traitement Automatique de la Langue Naturelle (TALN '09)*, Senlis, France, 24–26 June 2009.

[BOU 11] BOUDIN F., HUET S., TORRES-MORENO J.-M., "A graph-based approach to cross-language multi-document summarization", *Polibits: Research Journal on Computer Science and Computer Engineering with Applications*, vol. 43, pp. 1–10, 2011.

[BOU 13] BOUDIN F., MORIN E., "Keyphrase extraction for N-best reranking in multi-sentence compression", *2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Atlanta, GA, pp. 298–305, June 2013.

[BRA 95] BRANDOW R., MITZE K., RAU L., "Automatic condensation of electronic publications by sentence selection", *Information Processing and Management*, vol. 31, no. 5, pp. 675–685, September 1995.

[BRI 98] BRIN S., PAGE L., "The anatomy of a large-scale hypertextual web search engine", *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[BUY 02] BUYUKKOKTEN O., KALJUVEE O., GARCIA-MOLINA H., *et al.*, "Efficient web browsing on handheld devices using page and form summarization", *Transaction on Information Systems (TOIS)*, vol. 20, no. 1, pp. 82–115, January 2002.

[CAB 01] CABRÉ M.T., ESTOPÀ R., VIVALDI J., "Automatic term detection: a review of current systems", BOURIGAULT D., JACQUEMIN C., L'HOMME M.-C., (eds.), *Recent Advances in Computational Terminology*, John Benjamins, Amsterdam, pp. 53–87, 2001.

[CAR 98] CARBONELL J.G., GOLDSTEIN J., "The use of MMR, diversity-based reranking for reordering documents and producing summaries", MOFFAT A., ZOBEL J. (eds.), *ACM Special Interest Group on Information Retrieval (SIGIR '98)*, Melbourne, Australia, pp. 335–336, 1998.

[CAR 07] CARENINI G., NG R.T., ZHOU X., "Summarizing email conversations with clue words", WILLIAMSON C.L., ZURKO M.E., PATEL-SCHNEIDER P.F., *et al.*, (eds.), *WWW*, ACM, pp. 91–100, 2007.

[CEY 09] CEYLAN H., MIHALCEA R., "The decomposition of human-written book summaries", *10th International Conference Computational Linguistics and Intelligent Text Processing (CICLing '09)*, Mexico, Springer-Verlag, Berlin, pp. 582–593, 1–7 March 2009.

[CHA 06] CHARTIER S., BOUKADOUM M., "A sequential dynamic heteroassociative memory for multi step pattern recognition and one-to-many association", *IEEE Transactions on Neural Networks*, vol. 17, pp. 59–68, 2006.

[CHA 11] CHANG C.-C., LIN C.-J., "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[CHA 11] CHAKRABARTI D., PUNERA K., "Event summarization using tweets", *5th International Conference on Weblogs and Social Media (ICWSM)*, Association for the Advancement of Artificial Intelligence, 2011.

[CHI 09] CHIEZE E., FARZINDAR A., LAPALME G., "*An Automatic System for Summarization and Information Extraction of Legal Information*", Springer, pp. 1–20, 2009.

[CHU 00] CHUANG W.T., YANG J., "Text summarization by sentence segment extraction using machine learning algorithms", *4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications (PAKDD '00)*, Springer-Verlag, London, pp. 454–457, 2000.

[CLA 06a] CLARKE J., LAPATA M., "Constraint-based sentence compression: an integer programming approach", *COLING/ACL '06 on Main Conference Poster Sessions*, Sydney, Australia, pp. 144–151, 2006.

[CLA 06b] CLARKE J., LAPATA M., "Models for sentence compression: a comparison across domains, training requirements and evaluation measures", *21st ICCL and the 44th Meeting of the ACL*, Association for Computational Linguistics, Sydney, Australia, pp. 377–384, 2006.

[CLE 83] CLEVELAND D.B., CLEVELAND A.D., *Introduction to Indexing and Abstracting*, Libraries Unlimited, Littleton, CO, 1983.

[COH 09] COHN T., LAPATA M., "Sentence compression as tree transduction", *Journal of Artificial Intelligence Research*, vol. 34, pp. 637–674, 2009.

[COL 03] COLLINS M., "Head-driven statistical models for natural language parsing", *Computational Linguistics*, vol. 29, no. 4, pp. 589–637, 2003.

[CON 01] CONROY J.M., O'LEARY D.P., "Text summarization via hidden Markov models", *24th International Conference on Research and Development in Information Retrieval (SIGIR '01)*, ACM, New Orleans, LA, pp. 406–407, 2001.

[COP 04] COPECK T., SZPAKOWICZ S., "Vocabulary agreement among model summaries and source documents", *Document Understanding Conference (DUC '04)*, Boston, MA, 2004.

[COR 09] CORDEIRO J., DIAS G., BRAZDIL P., "Unsupervised induction of sentence compression rules", *2009 Workshop on Language Generation and Summarisation (UCNLG+Sum'09)*, ACL, Suntec, Singapore, pp. 15–22, 2009.

[CRE 93] CREMMINS E., "Valuable and meaningful text summarization in thoughts, words, and deeds", ENDRES-NIGGEMEYER B., HOBBS J., SPÄRCK-JONES K. (eds.), *Workshop on Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, 1993.

[CRE 96] CREMMINS E., *The Art of Abstracting*, 2nd ed., Book News, Portland, 1996.

[CUN 05] DA CUNHA I., WANNER L., "Towards the automatic summarization of medical articles in spanish: integration of textual, lexical, discursive and syntactic criteria", *Workshop Crossing Barriers in Text Summarization Research*, Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2005.

[CUN 07a] DA CUNHA I., FERNÁNDEZ S., VELÁZQUEZ-MORALES P., *et al.*, "A new hybrid summarizer based on vector space model, statistical physics and linguistics", *6th Mexican International Conference on Advances in Artificial Intelligence (MICAI '07)*, Springer-Verlag, Aguascalientes, Mexico, pp. 872–882, 2007.

[CUN 07b] DA CUNHA I., WANNER L., CABRÉ M.T., "Summarization of specialized discourse: the case of medical articles in Spanish", *Terminology*, vol. 13, no. 2, pp. 249–286, 2007.

[CUN 08] DA CUNHA I., Hacia un modelo lingüístico de resumen automático de artículos médicos en español, PhD Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[CUN 09] DA CUNHA I., TORRES-MORENO J.-M., VELÁZQUEZ P., *et al.*, “Un algoritmo lingüístico-estadístico para resumen automático de textos especializados”, *Linguamática*, vol. 2, pp. 67–79, 2009.

[CUN 11] CUNNINGHAM H., *et al.*, “Text processing with GATE (Version 6)”, University of Sheffield Department of Computer Science, 15 April 2011.

[CUN 12] DA CUNHA I., SANJUAN E., TORRES-MORENO J.-M., *et al.*, “DiSeg 1.0: the first system for Spanish discourse segmentation”, *Expert Systems With Applications*, vol. 39, no. 2, pp. 1671–1678, 2012.

[DAM 02] DAMIANOS L., WOHLEVER S., PONTE J., *et al.*, “Real users, real data, real problems: the MiTAP system for monitoring bio events”, *2nd International Conference on Human Language Technology Research*, HLT ’02, Morgan Kaufmann Publishers Inc., San Francisco, CA, pp. 357–362, 2002.

[DAN 06] DANG H.T., “DUC 2005: evaluation of question-focused summarization systems”, *Workshop on Task-Focused Summarization and Question Answering (SumQA ’06)*, ACL, Sydney, Australia, pp. 48–55, 2006.

[DAN 07] DANG H.T., “Overview of DUC 2007”, *Document Understanding Conference (DUC ’07)*, Rochester, NY, 2007.

[DAN 08] DANG H.T., OWCZARZAK K., “Overview of the TAC 2008 update summarization task”, *Workshop Text Analysis Conference (TAC ’08)*, Gaithersburg, MD, 2008.

[DAU 02] DAUMÉ-III H., ECHIHABI A., MARCU D., *et al.*, “GLEANS: a generator of logical extracts and abstracts for nice summaries”, *Document Understanding Conference (DUC ’02)*, Philadelphia, PA, 2002.

[DAU 05] DAUMÉ III H., MARCU D., “A Bayesian model for supervised clustering with the Dirichlet process prior”, *Journal of Machine Learning Research*, vol. 6, pp. 1551–1577, December 2005.

[DAU 06] DAUMÉ III H., Practical structured learning techniques for natural language processing, PhD Thesis, University of Southern California, Los Angeles, CA, 2006.

[DE 08] DE MARNEFFE M.-C., MANNING C.D., “The Stanford typed dependencies representation”, *COLING’08: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation (CrossParser ’08)*, Association for Computational Linguistics, Stroudsburg, PA, pp. 1–8, 2008.

[DEE 90] DEERWESTER S., DUMAIS S.T., FURNAS G.W., *et al.*, “Indexing by latent semantic analysis”, *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[DEJ 79] DEJONG G.F., Skimming stories in real time: an experiment in integrated understanding, PhD Thesis, Department of Computer Science, Yale University, New Haven, CT, 1979.

[DEJ 82] DEJONG G.F., “An overview of the FRUMP system”, LEHNERT W.G., RINGLE M.H. (eds.), *Strategies for Natural Language Processing*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 149–176, 1982.

[DIJ 80] VAN DIJK T.A., *Text and Context: Explorations in the Semantics and Pragmatics of Discourse*, Longman, London, 1980.

[DOD 02] DODDINGTON G., “Automatic evaluation of machine translation quality using N-gram co-occurrence statistics”, *Human Language Technology Conference (HLT ’02)*, San Diego, CA, 2002.

[DON 00] DONAWAY R.L., DRUMMEY K.W., MATHER L.A., “A comparison of rankings produced by summarization evaluation measures”, *NAACL Workshop on Automatic Summarization*, Seattle, WA, pp. 69–78, 2000.

[DOR 05] DORR B., MONZ C., PRESIDENT S., *et al.*, “A methodology for extrinsic evaluation of text summarization: does ROUGE correlate?”, *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, ACL, Ann Arbor, MI, pp. 1–8, June 2005.

[DOY 61] DOYLE L.B., “Semantic road maps for literature searchers”, *Journal of the Association for Computing Machinery*, vol. 8, no. 4, pp. 553–578, 1961.

[DOY 62] DOYLE L.B., “Indexing and abstracting by association”, *American Documentation*, vol. 13, no. 4, pp. 378–390, 1962,

[DUN 93] DUNNING T., “Accurate methods for the statistics of surprise and coincidence”, *Computational Linguistics*, vol. 19, no. 1, pp. 61–74, March 1993.

[EDM 61] EDMUNDSON H.P., WYLLYS R.E., “Automatic abstracting and indexing – survey and recommendations”, *Communications ACM*, vol. 4, no. 5, ACM, New York, pp. 226–234, May 1961.

[EDM 69] EDMUNDSON H.P., “New methods in automatic extraction”, *Journal of the Association for Computing Machinery*, vol. 16, no. 2, pp. 264–285, 1969.

[END 95a] ENDRES-NIGGEMEYER B., MAIER E., SIGEL A., “How to implement a naturalistic model of abstracting: four core working steps of an expert abstractor”, *Information Processing and Management*, vol. 31, no. 5, pp. 631–674, September 1995.

[END 95b] ENDRES-NIGGEMEYER B., NEUGEBAUER E., “Professional summarising: no cognitive simulation without observation”, *International Conference in Cognitive Science*, San Sebastian, Spain, 2–6 May 1995.

[END 98] ENDRES-NIGGEMEYER B., *Summarizing Information*, Springer, 1998.

[ERK 04] ERKAN G., RADEV D.R., “LexRank: graph-based lexical centrality as salience in text summarization”, *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.

[EVA 01] EVANS D.K., KLAVANS J.L., MCKEOWN K.R., “Columbia newsblaster: multilingual news summarization on the web”, *Document Understanding Conference (DUC ’01)*, New Orleans, LA, 2001.

[EVA 05] EVANS D.K., MCKEOWN K.R., “Identifying similarities and differences across English and Arabic news”, *International Conference on Intelligence Analysis*, McLean, VA, pp. 23–30, 2005.

[FAR 04a] FARZINDAR A., LAPALME G., DESCLÉS J.P., “Résumé de textes juridiques par identification de leur structure thématique”, *Traitement Automatique des Langues (TAL), Numéro spécial sur: Le résumé automatique de texte: solutions et perspectives*, vol. 45, no. 1, pp. 39–64, 2004.

[FAR 04b] FARZINDAR A., LAPALME G., “Legal text summarization by exploration of the thematic structures and argumentative roles”, *Text Summarization Branches Out Conference,* ACL, pp. 27–38, 2004.

[FAR 05] FARZINDAR A., ROZON F., LAPALME G., “CATS a topic-oriented multi-document summarization system”, *Document Understanding Conference (DUC ’05)*, Vancouver, Canada, October 2005.

[FAT 04] FATMA J.K., MAHER J., LAMIA B.H., *et al.*, “Summarization at LARIS laboratory”, *Document Understanding Conference (DUC ’04)*, Boston, MA, 2004.

[FAY 85] FAYOL M., *Le récit et sa construction. Une approche de psychologie cognitive*, Delachaux et Niestlé, Paris, 1985.

[FEL 98] FELLBAUM C., *WordNet. An Electronic Lexical Database*, MIT Press, Cambridge, 1998.

[FEL 07] FELDMAN R., SANGER J., *The Text Mining Bandbook : Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press, New York, 2007.

[FER 07] FERNÁNDEZ S., SANJUAN E., TORRES-MORENO J.-M., "Textual energy of associative memories: performants applications of Enertex algorithm in text summarization and topic segmentation", *Mexican International Conference on Artificial Intelligence (MICAI '07)*, Springer-Verlag, Aguascalientes, Mexico, pp. 861–871, 2007.

[FER 08a] FERNÁNDEZ S., SANJUAN E., TORRES-MORENO J.-M., "Enertex: un système basé sur l'énergie textuelle", *Traitement Automatique des Langues Naturelles (TALN '08)*, Avignon, France, pp. 99–108, 2008.

[FER 08b] FERNÁNDEZ S., VELÁZQUEZ P., MANDIN S., *et al.*, "Les systèmes de résumé automatique sont-ils vraiment des mauvais élèves?", *Journées internationales d'Analyse statistique des Données Textuelles (JADT '08)*, Lyon, France, 2008.

[FER 09a] FERNÁNDEZ S., SANJUAN E., TORRES-MORENO J.-M., "Résumés de texte par extraction de phrases, algorithmes de graphe et énergie textuelle", *XVI^{es} rencontres de la Société francophone de classification*, Grenoble, France, pp. 101–104, 2009.

[FER 09b] FERNÁNDEZ S., TORRES-MORENO J.-M., "Une approche exploratoire de compression automatique de phrases basée sur des critères thermodynamiques", *Traitement Automatique de la Langue Naturelle (TALN'09)*, Senlis, France, 24–26 June 2009.

[FIL 10] FILIPPOVA K., "Multi-sentence compression: finding shortest paths in word graphs", *International Conference on Computational Linguistics (COLING '10)*, Beijing, China, 2010.

[FLU 99] FLUHR C., FREDERKING R.E., OARD D., *et al.*, "Chapter 3 : multilingual (or cross-lingual) information retrieval", *Linguistica Computazionale*, vol. XIV-XV, Insituti Editoriali e Poligrafici Internazionali, Pisa, 1999.

[FOL 98] FOLTZ P.W., KINTSCH W., LANDAUER T.K., "The measurement of textual coherence with latent semantic analysis", *Discourse Processes*, vol. 25, nos. 2–3, pp. 285–307, 1998.

[FRI 04] FRIBURGER N., MAUREL D., "Finite-state transducer cascade to extract named entities in texts", *Theoretical Computer Science*, vol. 313, pp. 94–104, 2004.

[FUE 04] FUENTES M., GONZÁLEZ E., RODRÍGUEZ H., "Resumidor de noticies en catala del projecte hermes", *II Congres d'Enginyeria en Llengua Catalana (CELC '04)*, Andorre, 2004.

[FUE 08] FUENTES FORT M., A flexible multitask summarizer for documents from different media, domain, and language, PhD Thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain, 2008.

[GAG 06] GAGNON M., SYLVA L.D., "Text compression by syntactic pruning", LAMONTAGNE L., MARCHAND M. (eds.), *Canadian Conference on Artificial Intelligence (CCAI '06)*, vol. 4013, Springer-Verlag, Québec, Canada, pp. 312–323, 2006.

[GAI 01] GAIZAUSKAS R., HERRING P., OAKES M., *et al.*, "Intelligent access to text integrating information extraction technology into text browsers", *Human Language Technology Conference (HLT '01)*, San Diego, CA, pp. 189–193, 2001.

[GAL 06] GALLEY M., "Automatic summarization of conversational multi-party speech", *21st National Conference on Artificial intelligence (AAAI '06)*, AAAI Press, Boston, MA, pp. 1914–1915, 2006.

[GAL 07] GALLEY M., MCKEOWN K., "Lexicalized Markov grammars for sentence compression", *HLT-NAACL*, pp. 180–187, 2007.

[GEL 99] GELBUKH A., SIDOROV G., GUZMÁN-ARENAS A., "A method of describing document contents through topic selection", *International Symposium on String Processing and Information Retriveal (SPIRE/CRIWG '99)*, IEEE Computer Society Press, Cancun, Mexico, pp. 73–80, 1999.

[GEN 09a] GENEST P.-E., LAPALME G., NERIMA L., *et al.*, "A symbolic summarizer with 2 steps of sentence selection for TAC 2009", *Workshop Text Analysis Conference (TAC'09)*, Gaithersburg, MD, 2009.

[GEN 09b] GENEST P.-E., Système symbolique de création de résumés de mise à jour, Master's Thesis, DIRO, Faculté des arts et des sciences, University of Montreal, QC, Canada, 2009.

[GEN 10] GENEST P.-E., LAPALME G., YOUSFI-MONOD M., "Jusqu'où peut-on aller avec des méthodes par extraction pour la rédaction de résumés?", *Traitement Automatique des Langagues Naturelles (TALN '10)*, Montreal, QC, Canada, 19–23 July 2010.

[GEN 12] GENEST P.-E., LAPALME G., "Fully abstractive approach to guided summarization", *Proceedings of the 50th Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea, pp. 354–358, 8–14 July 2012.

[GIA 13] GIANNAKOPOULOS G., "Multi-document multilingual summarization and evaluation tracks in ACL 2013 multiLing workshop", *MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, Association for Computational Linguistics, Sofia, Bulgaria, pp. 20–28, August 2013.

[GIL 08] GILLICK D., FAVRE B., HAKKANI-TUR D., "The ICSI summarization system at TAC 2008", *Text Analysis Conference (TAC '08)*, Gaithersburg, MD, 2008.

[GIL 09] GILLICK D., FAVRE B., HAKKANI-TUR D., *et al.*, "The ICSI/UTD summarization system at TAC 2009", *Text Analysis Conference (TAC '09)*, Gaithersburg, MD, 2009.

[GIL 11] GILLICK D., The elements of automatic summarization, PhD Thesis, EECS Department, University of California, Berkeley, CA, May 2011.

[GÖK 95] GÖKÇAY D., GÖKÇAY E., "Generating titles for paragraphs using statistically extracted keywords and phrases", *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3174–3179, 1995.

[GOL 99] GOLDSTEIN J., KANTROWITZ M., MITTAL V., *et al.*, "Summarizing text documents: sentence selection and evaluation metrics", *22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)*, ACM, Berkeley, CA, pp. 121–128, 15–19 August 1999.

[GOL 00] GOLDSTEIN J., MITTAL V.O., CARBONELL J., *et al.*, "Multi-document summarization by sentence extraction", HAHN U., LIN C.Y., MANI I., *et al.*, (eds.), *Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, Seattle, WA, April 2000.

[GON 01] GONG Y., LIU X., "Generic text summarization using relevance measure and latent semantic analysis", *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01*, ACM, New York, NY, pp. 19–25, 2001.

[GRE 98] GREFENSTETTE G., “Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind”, *AAAI Spring Symposium on Intelligent Text summarization (Working notes)*, Stanford University, CA, pp. 111–118, 1998.

[HAR 03] HARABAGIU S.M., LACATUSU V.F., MAIORANO S.J., “Multi-document summaries based on semantic redundancy”, RUSSELL I., HALLER S.M., (eds.), *16th International Florida Artificial Intelligence Research Society Conference (FLAIRS ’03)*, AAAI Press, St. Augustine, FL, pp. 387–391, 12–14 May 2003.

[HAT 01] HATZIVASSILOGLOU V., KLAVANS J.L., HOLCOMBE M.L., *et al.*, “SIMFINDER: a flexible clustering tool for summarization”, *Workshop on Automatic Summarization (NAACL ’01)*, Pittsburgh, PA, pp. 41–49, 2001.

[HER 91] HERTZ J., KROGH A., PALMER R., *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, 1991.

[HIR 03] HIRAO T., SUZUKI J., ISOZAKI H., *et al.*, “NTT’s multiple document summarization system for DUC 2003”, *Document Understanding Conference (DUC ’03)*, Edmonton, Canada, 2003.

[HOP 82] HOPFIELD J., “Neural networks and physical systems with emergent collective computational abilities”, *National Academy of Sciences of the USA*, vol. 9, pp. 2554–2558, 1982.

[HOR 02] HORI C., FURUI S., MALKIN R., *et al.*, “Automatic speech summarization applied to English broadcast news speech”, *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’02)*, vol. 1, Orlando, FL, pp. 9–12, 2002.

[HOR 04] HORI C., FURUI S., “Speech summarization: an approach through word extraction and a method for evaluation”, *IEICE Transactions on Information and Systems (Institute of Electronics, Informatics and Communication Engineering)*, vol. 87, no. 1, pp. 15–25, January 2004.

[HOV 98] HOVY E., LIN C.-Y., “Automated text summarization and the SUMMARIST system”, TIPSTER ’98, ACL, Baltimore, MD, pp. 197–214, 1998.

[HOV 99] HOVY E., LIN C.-Y., “Automated text summarization in SUMMARIST”, MANI I., MAYBURY M.T. (eds.), *Advances in Automatic Text Summarization*, MIT Press, Cambridge, pp. 81–94, 1999.

[HOV 05] HOVY E., “Automated text summarization”, MITKOV R. ed., *The Oxford Handbook of Computational Linguistics*, Oxford University Press, Oxford, pp. 583–598, 2005.

[HOV 06] HOVY E., LIN C.-Y., ZHOU L., *et al.*, "Automated summarization evaluation with basic elements", *5th International Conference on Language Resources and Evaluation (LREC '06)*, Genoa, Italy, 24–26 May 2006.

[HU 07] HU M., SUN A., LIM E.-P., "Comments-oriented blog summarization by sentence extraction", *16th ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, ACM, New York, NY, pp. 901–904, 2007.

[IPM 95] IPM, "Special issue on automatic summarizing", *Information Processing and Management*, vol. 31, no. 3, pp. 625–784, 1995.

[ISH 02] ISHIKAWA K., ANDO S.I., DOI S.I., *et al.*, "Trainable automatic text summarization using segmentation of sentence", *Workshop NII Test Collection for IR Systems (NTCIR 3 TSC)*, Tokyo, Japan, 2002.

[ISR 96] ISRAEL D., KAMEYAMA M., STICKEL M., *et al.*, "FASTUS: a cascaded finite-state transducer for extracting information from natural-language text", ROCHE E., SCHABES Y. (eds.), *Finite State Devices for Natural Language Processing*, MIT Press, Cambridge, pp. 383–406, 1996.

[JAR 89] JARO, M. A., "Advances in record linkage methodology as applied to the 1985 census of Tampa Florida", *American Statistical Association*, vol. 84, no. 406, pp. 414– 20, 1989.

[JIN 98] JING H., MCKEOWN K.R., "Combining multiple, large-scale resources in a reusable lexicon for natural language generation", *17th International Conference on Computational Linguistics – 36th Meeting of the Association for Computational Linguistics (COLING-ACL '98)*, Montreal, QC, Canada, Université de Montréal, pp. 607–613, August 1998.

[JIN 99] JING H., MCKEOWN K.R., "The decomposition of human-written summary sentences", *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, ACM, New York, NY, pp. 129–136, 1999.

[JIN 00a] JING H., "Sentence reduction for automatic text summarization", *6th Conference on Applied Natural Language Processing (ANLP'00)*, Association for Computational Linguistics, Seattle, WA, pp. 310–315, 2000.

[JIN 00b] JING H., MCKEOWN K.R., "Cut and paste based text summarization", *1st North American Chapter of the Association for Computational Linguistics*, Seattle, WA, pp. 178–185, 2000.

[JIN 02] JING H., “Using hidden Markov modeling to decompose human-written summaries”, *Computational Linguistics*, vol. 4, no. 28, pp. 527–543, 2002.

[JIN 03] JIN R., Statistical approaches toward title generation, PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, 2003.

[JOA 08] JOAN A., VIVALDI J., LORENTE M., “Turning a term extractor into a new domain: first experiences”, *LREC’08*, ELRA, Marrakech, Morocco, 28–30 May 2008.

[JOH 84] JOHN F.S., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984.

[JOH 02] JOHNSON D.B., ZOU Q., DIONISIO J.D., *et al.*, “Modeling medical content for automated summarization”, *Annals of the New York Academy of Sciences*, vol. 980, pp. 247–258, 2002.

[KAB 13] KABADJOV M., STEINBERGER J., STEINBERGER R., “Multilingual statistical news summarization”, POIBEAU T., SAGGION H., PISKORSKI J. *et al.*, (eds.), *Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, Springer, Berlin, vol. 2013, pp. 229–252, 2013.

[KAN 87] KANTER L., SOMPOLINSKY H., “Associative recall of memory without errors”, *Physical Review A*, vol. 35, pp. 380–392, 1987.

[KAN 01] KAN M.Y., MCKEOWN K.R., KLAVANS J.L., “Domain-specific informative and indicative summarization for information retrieval”, *Document Understanding Conference (DUC ’01)*, NIST, New Orleans, LA, pp. 19–26, 2001.

[KIP 08] KIPPER K., KORHONEN A., RYANT N., *et al.*, *A Large-Scale Classification of English Verbs, Language Resources and Evaluation*, Springer, Netherland, vol. 42, no. 1, pp. 21–40, 2008.

[KIS 06] KISS T., STRUNK J., “Unsupervised multilingual sentence boundary detection”, *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.

[KNI 00] KNIGHT K., MARCU D., “Statistics-based summarization – step one: sentence compression”, *17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, Austin, TX, pp. 703–710, 2000.

[KNI 02] KNIGHT K., MARCU D., “Summarization beyond sentence extraction: a probabilistic approach to sentence compression”, *Artificial Intelligence*, vol. 139, no. 1, pp. 91–107, July 2002.

[KOS 88] KOSKO B., "Bidirectional associative memories", *IEEE Transactions on Systems Man, Cybernetics*, vol. 18, pp. 49–60, 1988.

[KUL 51] KULLBACK S., LEIBLER R., "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[KUP 95] KUPIEC J., PEDERSEN J., CHEN F., "A trainable document summarizer", *18th Conference ACM Special Interest Group on Information Retrieval (SIGIR'95)*, Seattle, WA, ACM Press, New York, pp. 68–73, 1995.

[LAL 02] LAL P., RUGER S., "Extract-based summarization with simplification", *Document Understand Conference (DUC'02)*, NIST, Philadelphie, PA, 2002.

[LAN 98] LANDAUER T.K., FOLTZ P.W., LAHAM D., "An introduction to latent semantic analysis", *Discourse Processes*, vol. 25, no. 2, pp. 259–284, 1998.

[LAP 13] LAPALME G., "Natural language generation and summarization at RALI", *14th European Workshop on Natural Language Generation*, Association for Computational Linguistics, Sofia, Bulgaria, pp. 92–93, August 2013.

[LEB 04] LEBART L., "Validité des visualisations de données textuelles", *Journéés d'Analyse Textuelle de Documents (JADT '04)*, Louvain-La-Neuve, Belgium, pp. 708–715, 2004.

[LEH 95] LEHMAM A., Le résumé des textes techniques et scientifiques, aspects linguistiques et computationnels, PhD Thesis, University of Nancy 2, France, 1995.

[LEM 05] LEMAIRE B., MANDIN S., DESSUS P., *et al.*, "Computational cognitive models of summarization assessment skills", BARA B.G., BARSALOU L., BUCCIARELLI M. (eds.), *27th Conference CogSci'05*, Stresa, Italy, pp. 1266–1271, 2005.

[LEN 02] LENCI A., BARTOLINI R., CALZOLARI N., *et al.*, "Multilingual summarization by integrating linguistic resources in the MLIS-MUSI project", *3rd International Conference on Language Resources and Evaluation (LREC '02)*, Canary Islands, Spain, pp. 1464–1471, 29–31 May 2002.

[LIN 91] LIN J., "Divergence measures based on the Shannon entropy", *IEEE Transactions on Information Theory*, vol. 37, pp. 145–151, 1991.

[LIN 95] LIN C.-Y., “Knowledge-based automatic topic identification”, *33rd Meeting of the Association for Computational Linguistics (ACL ’95)*, Morgan Kaufmann, Cambridge, MA, pp. 308–310, 26–30 June 1995.

[LIN 97] LIN C.-Y., HOVY E., “Identifying topics by position”, *5th Conference on Applied Natural Language Processing (ANLC ’97)*, ACL, Washington, DC, pp. 283–290, 1997.

[LIN 98] LIN C.-Y., “Assembly of topic extraction modules in SUMMARIST”, *AAAI Spring Symposium on Intelligent Text Summarization*, Association for the Advancement of Artificial Intelligence, Stanford University, CA, pp. 53–59, 1998.

[LIN 99] LIN C.-Y., “Training a selection function for extraction”, *8th International Conference on Information and Knowledge Management (CIKM ’99)*, ACM, Kansas City, MO, pp. 55–62, 1999.

[LIN 03a] LIN C.Y., “Improving summarization performance by sentence compression-a pilot study”, *6th International Workshop on Information Retrieval with Asian Languages*, Sapporo, Japan, pp. 1–8, 2003.

[LIN 03b] LIN C.-Y., HOVY E., “Automatic evaluation of summaries using N-gram co-occurrence statistics”, *2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL ’03)*, ACL, vol. 1, Edmonton, Canada, pp. 71–78, 2003.

[LIN 04] LIN C.-Y., “ROUGE: a package for automatic evaluation of summaries”, MOENS M.-F., SZPAKOWICZ S. (eds.), *Workshop Text Summarization Branches Out (ACL ’04)*, ACL, Barcelona, Spain, pp. 74–81, July 2004.

[LIN 06] LIN C.-Y., CAO G., GAO J., *et al.*, “An information-theoretic approach to automatic evaluation of summaries”, *Conference on Human Language Technology Conference of the North American Chapter*, ACL, Morristown, NJ, pp. 463–470, 2006.

[LIT 10a] LITVAK M., LAST M., FRIEDMAN M., “A new approach to improving multilingual summarization using a genetic algorithm”, *48th Meeting of the Association for Computational Linguistics (ACL ’10)*, ACL, Uppsala, Sweden, pp. 927–936, 11–16 July 2010.

[LIT 10b] LITVAK M., LAST M., KISILEVICH S., *et al.*, “Towards multilingual summarization: a comparative analysis of sentence extraction methods on English and Hebrew corpora”, *4th Workshop on Cross Lingual Information Access (COLING ’10)*, Beijing, China, pp. 61–69, August 2010.

[LIT 12] LITVAK M., VANETIK N., “Polytope model for extractive summarization”, FRED A.L.N., FILIPE J., FRED A.L.N. *et al.*, (eds.), *KDIR*, SciTePress, pp. 281–286, 2012.

[LIT 13] LITVAK M., VANETIK N., “Mining the gaps: towards polynomial summarization”, *6th International Joint Conference on Natural Language Processing*, Asian Federation of Natural Language Processing, Nagoya, Japan, pp. 655–660, October 2013.

[LIT 14] LITVAK M., VANETIK N., “Multi-document summarization using tensor decomposition”, *Computación y Sistemas (CYS)*, vol. 18, no. 3, 2014.

[LIU 09] LIU B., “Sentiment analysis”, *Keynote talk at the 5th Text Analytics Summit*, Boston, MA, 2009.

[LIU 12] LIU X., LI Y., WEI F., *et al.*, “Graph-based multi-tweet summarization using social signals”, *International Conference on Computational Linguistics (COLING ’12)*, pp. 1699–1714, 2012.

[LOP 02] LOPER E., BIRD S., “NLTK: the natural language toolkit”, *Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and computational linguistics (ACL-ETMTNLP ’02)*, ACL, Philadelphie, PA, pp. 63–70, 2002.

[LOU 08] LOUIS A., NENKOVA A., “Automatic summary evaluation without human models”, *First Text Analysis Conference (TAC’08)*, Gaithersburg, MD, 17–19 November 2008.

[LOU 09] LOUIS A., NENKOVA A., “Automatically evaluating content selection in summarization without human models”, *Conference on Empirical Methods in Natural Language Processing*, ACL, Singapore, pp. 306–314, 6–7 August 2009.

[LOU 10] DE LOUPY C., GUÉGAN M., AYACHE C., *et al.*, “A French human reference corpus for multi-document summarization and sentence compression”, *LREC’10*, ELRA, Valletta, Malta, 19–21 May 2010.

[LOU 13] LOUIS A., NENKOVA A., “Automatically assessing machine summary content without a gold standard”, *Computational Linguistics*, vol. 39, no. 2, pp. 267–300, 2013.

[LUH 58] LUHN H., “The automatic creation of literature abstracts”, *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[MAN 87] MANN W.C., THOMPSON S.A., *Rhetorical Structure Theory: A Theory of Text Organization*, University of Southern California, Information Sciences Institute, Marina del Rey, CA, 1987.

[MAN 88] MANN W., THOMPSON S., “Rhetorical structure theory : toward a functional theory of text organization”, *Text*, vol. 8, no. 3, pp. 243–281, 1988.

[MAN 98] MANI I., BLOEDORN E., “Machine learning of generic and user-focused summarization”, *AAAI ’98/IAAI ’98*, Madison, WI, American Association for Artificial Intelligence, Menlo Park, CA, pp. 820–826, 1998.

[MAN 99a] MANI I., MAYBURI M., *Advances in Automatic Text Summarization*, MIT Press, Cambridge, 1999.

[MAN 99b] MANNING C.D., SCHÜTZE H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, 1999.

[MAN 00] MANI I., WILSON G., “Robust temporal processing of news”, *38th Association for Computational Linguistics*, Morristown, NJ, pp. 69–76, 2000.

[MAN 01] MANI I., *Automatic Summarization*, John Benjamins, Amsterdam, 2001.

[MAN 02] MANI I., KLEIN G., HOUSE D., *et al.*, “SUMMAC: a text summarization evaluation”, *Natural Language Engineering*, vol. 8, no. 1, pp. 43–68, March 2002.

[MAN 08] MANNING C.D., RAGHAVAN P., SCHÜTZE H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[MAN 09] MANNA S., PETRES Z., GEDEON T., “Tensor term indexing: an application of HOSVD for document summarization”, *4th International Symposium on Computational Intelligence and Intelligent Informatics*, pp. 135–141, 2009.

[MAR 98] MARCU D., The rhetorical parsing, summarization, and generation of natural language texts, PhD Thesis, Computer Science, University of Toronto, Toronto, Canada, 1998.

[MAR 99] MARCU D., “The automatic construction of large-scale corpora for summarization research”, *22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’99)*, Berkeley, CA, pp. 137–144, 1999.

[MAR 00a] MARCU D., “The rhetorical parsing of unrestricted texts: a surface-based approach”, *Computational Linguistics*, vol. 26, no. 3, pp. 395–448, 2000.

[MAR 00b] MARCU D., *The Theory and Practice of Discourse Parsing Summarization*, MIT Press, Cambridge, 2000.

[MAT 72] MATHIS B.A., Techniques for the evaluation and improvement of computer-produced abstracts, Report, Computer and Information Science Research Center, Ohio State University, Columbus, OH, 1972.

[MAT 03] MATEO P.L., GONZÁLEZ J.C., VILLENA J., *et al.*, “Un sistema para resumen automático de textos en castellano”, *Procesamiento del lenguaje natural*, vol. 31, pp. 29–36, September 2003.

[MAY 95] MAYBURY M.T., “Generating summaries from event data”, *Information Processing and Management*, vol. 31, no. 5, pp. 735–751, 1995.

[MAY 02] MAYNARD D., TABLAN V., CUNNINGHAM H., *et al*, “Architectural elements of language engineering robustness”, *Natural Language Engineering*, vol. 8, nos. 2–3, pp. 257–274, 2002.

[MCK 85] MCKEOWN K.R., “Discourse strategies for generating natural-language text”, *Artificial Intelligence*, vol. 27, pp. 1–41, 1985.

[MCK 93] MCKEOWN K., “Generating the complex sentences of summaries using syntactic and lexical constraints: two applications”, ENDRES-NIGGEMEYER B., HOBBS J., SPÄRCK-JONES K. (eds.), *Dagstuhl Seminar Report (9350), Workshop on Summarising Text for Intelligent Communication. Dagstuhl*, Schloss Dagstuhl, Germany, 1993.

[MCK 95a] MCKEOWN K., RADEV D.R., “Generating summaries of multiple news articles”, *18th Conference ACM Special Interest Group on Information Retrieval (SIGIR ’95)*, Seattle, WA, ACM Press, New York, pp. 74–82, 1995.

[MCK 95b] MCKEOWN K., ROBIN J., KUKICH K., “Generating concise natural language summaries”, *Information Processing and Management*, vol. 31, no. 5, pp. 703–733, 1995.

[MCK 99] MCKEOWN K.R., KLAVANS J.L., HATZIVASSILOGLOU V.,*et al.*, “Towards multidocument summarization by reformulation: progress and prospects”, *16th National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference (AAAI ’99/IAAI ’99)*, American Association for Artificial Intelligence, Orlando, FL, pp. 453–460, 1999.

[MCK 01] MCKEOWN K.R., HATZIVASSILOGLOU V., BARZILAY R., *et al.*, “Columbia multi-document summarization: approach and evaluation”, *Document Understanding Conference (DUC ’01)*, New Orleans, LA, 2001.

[MCK 02] McKEOWN K.R., BARZILAY R., EVANS D., *et al.*, "Tracking and summarizing news on a daily basis with Columbia's newsblaster", *2nd International Conference on Human Language Technology Research (HLT '02)*, Morgan Kaufmann, San Diego, CA, pp. 280–285, 2002.

[MCK 03] McKEOWN K., BARZILAY R., CHEN J., *et al.*, "Columbia's newsblaster: new features and future directions", *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-Demonstrations'03)*, vol. 4, Edmonton, Canada, ACL, pp. 15–16, 2003.

[MCK 05] McKEOWN K., PASSONNEAU R.J., ELSON D.K., *et al.*, "Do summaries help? A task-based evaluation of multi-document summarization", *28th International Conference on Research and Development in Information Retrieval (SIGIR '05)*, Salvador, Brazil, ACM, pp. 210–217, 15–19 August 2005.

[MEL 88] MEL'CUK I., *Dependency Syntax: Theory and Practice*, State University Press of New York, Albany, NY, 1988.

[MEL 01] MEL'CUK I., *Communicative Organization in Natural Language. The Semantic-communicative Structure of Sentences*, John Benjamins, Amsterdam, 2001.

[MIH 04a] MIHALCEA R., TARAU P., "TextRank: bringing order into texts", *EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, July 2004.

[MIH 04b] MIHALCEA R., "Graph-based ranking algorithms for sentence extraction, applied to text summarization", *Interactive Poster and Demonstration Sessions (ACL '04)*, Barcelona, Spain, ACL, pp. 181–184, 21–26 July 2004.

[MIH 05a] MIHALCEA R., "Language independent extractive summarization", *43rd Meeting of the Association for Computational Linguistics (ACL '05)*, Ann Arbor, MI, ACL, pp. 49–52, 25–30 June 2005.

[MIH 05b] MIHALCEA R., TARAU P., "A language independent algorithm for single and multiple document summarization", *International Joint Conference on Natural Language Processing (IJCNLP '05)*, Jeju Island, South Korea, 2005.

[MIK 02] MIKHEEV A., "Periods, capitalized words, etc.", *Computational Linguistics*, vol. 28, no. 3, pp. 289–318, September 2002.

[MIL 95] MILLER G.A., "WordNet: a lexical database for English", *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[MIN 01] MINEL J.-L., DESCLÉS J.-P., CARTIER E., *et al.*, "Résumé automatique par filtrage sémantique d'informations dans des textes", *Traitement automatique de la Langue Naturelle*, vol. 20, no. 3, pp. 369–395, 2001.

[MIN 02] MINEL J.-L., *Filtrage sémantique : du résumé automatique à la fouille de textes*, Hermès-Lavoisier, Paris, 2002.

[MIR 13] MIRANDA-JIMÉNEZ S., GELBUKH A.F., SIDOROV G., "Summarizing conceptual graphs for automatic summarization task", *21th International Conference on Conceptual Structures (ICCS 2013)*, LNCS, Springer, vol. 7735, pp. 245–253, 2013.

[MIT 97] MITRA M., SINGHAL A., BUCKLEY C., "Automatic text summarization by paragraph extraction", MANI I., MAYBURY M. (eds.), *Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, pp. 39–46, July 1997.

[MOE 00] MOENS M.-F., *Automatic Indexing and Abstracting of Document Texts*, Kluwer Academic, Boston, MA, 2000.

[MOL 10a] MOLINA A., DA CUNHA I., TORRES-MORENO J.-M., *et al.*, "La compresión de frases: un recurso para la optimización de resumen automático de documentos", *Linguamática*, vol. 2, no. 3, pp. 13–27, 2010.

[MOL 10b] MOLINA A., SIERRA G., TORRES-MORENO J.-M., "La energía textual como medida de distancia en agrupamiento de definiciones", *Journées d'Analyse Statistique de Documents (JADT '10)*, Rome, Italy, 2010.

[MOL 13a] MOLINA A., Compression automatique de phrases: une étude vers la génération de résumés, PhD Thesis, Université d'Avignon et des Pays de Vaucluse, Avignon, France, 2013.

[MOL 13b] MOLINA A., SANJUAN E., TORRES-MORENO J.-M., "A turing test to evaluate a complex summarization task", *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, LNCS, Springer, Berlin, vol. 8138, pp. 75–80, 2013.

[MOL 13c] MOLINA A., TORRES-MORENO J.-M., SANJUAN E., *et al.*, "Discursive sentence compression", GELBUKH A. ed., *Computational Linguistics and Intelligent Text Processing*, Springer Berlin, LNCS, vol. 7817, pp. 394–407, 2013.

[MOR 99] MORRIS A.H., KASPER G.M., ADAMS D.A., "The effects and limitations of automated text condensing on reading comprehension performance", MANI I., MAYBURY M.T. (eds.), *Advances in Automatic Text Summarization*, pp. 305–323, MIT Press, Cambridge, 1999.

[MUR 01] MURESAN S., TZOUKERMANN E., KLAVANS J.L., "Combining linguistic and machine learning techniques for email summarization", *Workshop on Computational Natural Language Learning*, vol. 7 of *ConLL'01*, ACL, Toulouse, France, pp. 19:1–19:8, 2001.

[NAM 00] NAMER F., "Flemm, Un analyseur flexionnel du français à base de règles", *Traitement Automatique des Langues (TAL)*, vol. 41, no. 2, pp. 523–247, 2000.

[NAN 99] NANBA H., OKUMURA M., "Towards multi-paper summarization using reference information", *16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, Morgan Kaufmann, pp. 926–931, 31 July 1999.

[NEN 04] NENKOVA A., PASSONNEAU R.J., "Evaluating content selection in summarization: the pyramid method", *Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '04)*, Boston, MA, pp. 145–152, May 2004.

[NEN 07] NENKOVA A., PASSONNEAU R., MCKEOWN K., "The pyramid method: incorporating human content selection variation in summarization evaluation", *ACM Transactions on Speech Language Processing*, vol. 4, no. 2, pp. 1–23, 2007.

[NEN 11] NENKOVA A., MCKEOWN K., "Automatic summarization", *Foundations and Trends in Information Retrieval*, vol. 5, nos. 2–3, pp. 103–233, 2011.

[NET 00] NETO J.L., SANTOS A., KAESTNER C.A. *et al.*, "Generating text summaries through the relative importance of topics", *International Joint Conference, 7th Ibero-American Conference on AI: Advances in Artificial Intelligence (IBERAMIA-SBIA'00)*, Springer-Verlag, London, pp. 300–309, 2000.

[NET 02] NETO J.L., FREITAS A.A., KAESTNER C.A., "Automatic text summarization using a machine learning approach", *16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence (SBIA'02)*, Springer-Verlag, London, pp. 205–215, 2002.

[NEW 04] NEWMAN E., DORAN W., STOKES N., *et al.*, "Comparing redundancy removal techniques for multi-document summarisation", *2nd Starting AI Researchers Symposium (STAIRS '04)*, Valencia, Spain, pp. 223–228, August 2004.

[NOM 94] NOMOTO T., NITTA Y., "A grammatico-statistical approach to discourse partitioning", *15th ICCL*, Kyoto, Japan, pp. 1145–1150, 5–9 August 1994.

[OGD 99] OGDEN W., COWIE J., DAVIS M., *et al.*, "Keizai: an interactive cross-language text retrieval system", *Machine Translation Summit VII, Workshop on Machine Translation for Cross Language Information Retrieval*, Singapore, pp. 13–17, 1999.

[ONO 94] ONO K., SUMITA K., MIIKE S., "Abstract generation based on rhetorical structure extraction", *International Conference on Computational Linguistics (COLING'94)*, Association for Computational Linguistics, Kyoto, Japan, pp. 344–348, 1994.

[ORA 07] ORASAN C., EVANS R., "NP animacy identification for anaphora resolution", *Journal of Artificial Intelligence Results*, vol. 29, pp. 79–103, 2007.

[ORA 08] ORASAN C., CHIOREAN O.A., "Evaluation of a cross-lingual Romanian-English multi-document summariser", *LREC '08*, ELRA, Marrakech, Morocco, 28–30 May 2008.

[OVE 07] OVER P., DANG H., HARMAN D., "DUC in context", *Information Processing and Management*, vol. 43, no. 6, pp. 1506–1520, 2007.

[OWK 09] OWKZARZAK K., DANG H.T., "Evaluation of automatic summaries: metrics under varying data conditions", *UCNLG+Sum'09*, Suntec, Singapore, pp. 23–30, August 2009.

[PAG 98] PAGE L., BRIN S., MOTWANI R., *et al.*, The PageRank citation ranking: bringing order to the web, report, Stanford Digital Library Technologies Project, USA, 1998.

[PAI 90] PAICE C.D., "Constructing literature abstracts by computer: techniques and prospects", *Information Processing and Management*, vol. 26, no. 1, pp. 171–186, 1990.

[PAI 93] PAICE C.D., JONES P.A., "The identification of important concepts in highly structured technical papers", *16th International Conference on Research and Development in Information Retrieval (SIGIR '93)*, Pittsburgh, PA, ACM Press, New York, pp. 69–78, 1993.

[PAL 10] PALMER D.D., “Text pre-processing”, INDURKHYA N., DAMERAU F.J. (eds.), *Handbook of Natural Language Processing, Second Edition*, CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010.

[PAP 02] PAPINENI K., ROUKOS S., WARD T., *et al.*, “BLEU: a method for automatic evaluation of machine translation”, *40th Meeting of the Association for Computational Linguistics (ACL ’02)*, Philadelphia, PA, ACL, pp. 311–318, 6–12 July 2002.

[PAR 08] PARDO T., NUNES M., “On the development and evaluation of a Brazilian Portuguese discourse parser”, *Journal of Theoretical and Applied Computing*, vol. 15, no. 2, pp. 43–64, 2008.

[PLA 10] PLAZA L., STEVENSON M., DÍAZ A., “Improving summarization of biomedical documents using word sense disambiguation”, *Workshop on Biomedical Natural Language Processing*, Uppsala, Sweden, ACL, pp. 55–63, 15 July 2010.

[POI 11] POIBEAU T., *Traitement automatique du contenu textuel*, Hermès Lavoisier, 2011.

[POL 75] POLLOCK J., ZAMORA A., “Automatic abstracting research at chemical abstracts service”, *Journal of Chemical Information and Computer Sciences*, vol. 4, no. 15, pp. 226–233, 1975.

[POR 80] PORTER M.F., “An algorithm for suffix stripping”, *Program*, vol. 14, no. 3, pp. 130–137, July 1980.

[QUI 04] QUIRK C.B., “Training a sentence-level machine translation confidence measure”, *4th International Conference on Language Resources and Evaluation (LREC’04)*, Lisbon, Portugal, 24–28 May 2004.

[RAD 98] RADEV D.R., MCKEOWN K.R., “Generating natural language summaries from multiple on-line sources”, *Computational Linguistics*, vol. 24, no. 3, pp. 470–500, 1998.

[RAD 00] RADEV D.R., JING H., BUDZIKOWSKA M., “Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies”, *Workshop on Automatic Summarization (NAACL-ANLP-AutoSum’00)*, Seattle, WA, ACL, pp. 21–30, 2000.

[RAD 01a] RADEV D.R., BLAIR-GOLDENSOHN S., ZHANG Z., “Experiments in single and multi-document summarization using MEAD”, *1st Document Understanding Conference (DUC’01)*, New Orleans, LA, 2001.

[RAD 01b] RADEV D.R., BLAIR-GOLDENSOHN S., ZHANG Z., *et al.*, "Interactive, domain-independent identification and summarization of topically related news articles", *5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL '01)*, Darmstadt, Germany, Springer-Verlag, London, pp. 225–238, 2001.

[RAD 01c] RADEV D.R., FAN W., ZHANG Z., "WebInEssence: a personalized web-based multi-document summarization and recommendation system", *Workshop on Automatic Summarization (NAACL '01)*, Pittsburgh, PA, pp. 79–88, 2001.

[RAD 01d] RADEV D.R., GOLDENSOHN S.B., ZHANG Z., *et al.*, "NewsInEssence: a system for domain-independent, real-time news clustering and multi-document summarization", *Human Language Technology Conference*, San Diego, CA, 2001.

[RAD 02a] RADEV D., WINKEL A., TOPPER M., "Multi document centroid-based text summarization", *40th Meeting of the Association for Computational Linguistics (ACL '02), Demonstrations Session*, Philadelphie, PA, ACL, pp. 112–113, July 2002.

[RAD 02b] RADEV D.R., HOVY E., MCKEOWN K., "Introduction to the special issue on text summarization", *Computational Linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[RAD 03] RADEV D.R., TEUFEL S., SAGGION H., *et al.*, "Evaluation challenges in large-scale document summarization", *41st Meeting on Association for Computational Linguistics (ACL'03)*, Sapporo, Japan, ACL, pp. 375–382, 7–12 July 2003.

[RAD 04] RADEV D.R., JING H., STY M., *et al.*, "Centroid-based summarization of multiple documents", *Information Processing and Management*, vol. 40, no. 6, pp. 919–938, November 2004.

[RAM 04] RAMBOW O., SHRESTHA L., CHEN J., *et al.*, "Summarizing email threads", *HLT-NAACL Short Papers (HLT-NAACL-Short'04)*, Boston, MA, ACL, pp. 105–108, 2004.

[RAT 61] RATH G., RESNICK A., SAVAGE R., "The formation of abstracts by the selection of sentences. Part I: sentence selection by man and machines", *American Documentation*, vol. 12, no. 2, pp. 139–143, 1961.

[RAU 89] RAU L.F., JACOBS P.S., ZERNIK U., "Information extraction and text summarization using linguistic knowledge acquisition", *Information Processing and Management*, vol. 25, no. 4, pp. 419–428, 1989.

[RAU 94] RAU L.F., BRANDOW R., MITZE K., "Domain-independent summarization of news", *Summarizing Text for Intelligent Communication*, Dagstuhl, Germany, pp. 71–75, 1994.

[RAY 09] RAYBAUD S., LANGLOIS D., SMALI K., "Efficient combination of confidence mMeasures for machine translation", *10th Conference of the International Speech*, Brighton, UK, pp. 424–427, 2009.

[REE 07] REEVE L.H., HAN H., "The use of domain-specific concepts in biomedical text summarization", *Information Processing and Management*, vol. 43, no. 6, pp. 1765–1776,2007.

[REI 00] REITER E., DALE R., *Building Natural Language Generation Systems*, Cambridge University Press, New York, USA, 2000.

[RIE 03] RIEZLER S., KING T.H., CROUCH R., *et al.*, "Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar", *Conference of the North American Chapter of the ACL on Human Language Technology (NAACL'03)*, vol. 1, Edmonton, Canada, ACL, pp. 118–125, 2003.

[RIE 10] RIEDHAMMER K., FAVRE B., HAKKANI-TÜR D., "Long story short – global unsupervised models for keyphrase based meeting summarization", *Speech Communication*, vol. 52, no. 10, pp. 801–815, 2010.

[RIL 93] RILOFF E., "A corpus-based approach to domain-specific text summarisation: a proposal", ENDRES-NIGGEMEYER B., HOBBS J., SPÄRCK-JONES K., (eds.), *Workshop on Summarising Text for Intelligent Communication*, Dagstuhl Seminar Report (9350), Dagstuhl Seminar, Germany, pp. 69–84, 1993.

[ROG 01] ROGER S., "Belief revision on anaphora resolution", *Conference Computational Linguistics and Intelligent Text Processing (CICLing'01)*, vol. 2004/2001 of LNCS, Mexico, Springer-Verlag, Berlin, pp. 140–141, 2001.

[RUS 71] RUSH J.E., SALVADOR R., ZAMORA A., "Automatic abstracting and indexing. II. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria", *Journal of the American Society for Information Science*, vol. 22, no. 4, pp. 260–274, 1971.

[SAF 96] SAFIR K., "Semantic atoms of anaphora", *Natural Language & Linguistic Theory*, vol. 14, no. 3, pp. 545–589, 1996.

[SAG 00a] SAGGION H., LAPALME G., "Concept identification and presentation in the context of technical text summarization", *ANLP/NAACL Workshop on Automatic Summarization*, Seattle, WA, ACL, pp. 1–10, 2000.

[SAG 00b] SAGGION H., Génération automatique de résumés par analyse sélective, PhD Thesis, University of Montreal, QC, Canada, 2000.

[SAG 02a] SAGGION H., RADEV D., TEUFEL S., *et al.*, "Meta-evaluation of summaries in a cross-lingual environment using content-based metrics", *International Conference on Computational Linguistics (COLING'02)*, Taipei, Taiwan, pp. 849–855, August 2002.

[SAG 02b] SAGGION H., LAPALME G., "Generating indicative-informative summaries with SumUM", *Computational Linguistics*, vol. 28, no. 4, pp. 497–526, 2002.

[SAG 04] SAGGION H., GAIZAUSKAS R., "Multi-document summarization by cluster/profile relevance and redundancy removal", *Document Understanding Conference (DUC'04)*, Boston, MA, 2004.

[SAG 08] SAGGION, H., "SUMMA: A Robust and Adaptable Summarization Tool", *Traitement Automatique des Langues*, vol. 49, no. 2, pp. 103–125, 2008.

[SAG 10] SAGGION H., TORRES-MORENO J.-M., DA CUNHA I., *et al.*, "Multilingual summarization evaluation without human models", *23rd International Conference on Computational Linguistics (COLING'10)*, Beijing, China, ACL, pp. 1059–1067, 2010.

[SAG 14] SAGGION H., "Creating Summarization Systems with SUMMA", *LREC*, pp. 4157–4163, 2014.

[SAK 01] SAKAI T., SPÄRCK-JONES K., "Generic summaries for indexing in information retrieval", *ACM Special Interest Group on Information Retrieval (SIGIR'01): 24th International Conference on Research and Development in Information Retrieval*, New Orleans, LA, ACM, pp. 190–198, 2001.

[SAL 68] SALTON G., *Automatic Information Organization and Retrieval*, McGraw-Hill, New York, 1968.

[SAL 71] SALTON G., *The SMART Retrieval System – Experiments in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, 1971.

[SAL 83] SALTON G., MCGILL M., *Introduction to Modern Information Retrieval*, Computer Science Series, McGraw-Hill, New York, 1983.

[SAL 89] SALTON G., *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Boston, 1989.

[SAL 01] SALGUEIRO PARDO T.A., RINO MACHADO L.H., “A summary planner based on a three-level discourse model”, *6th Natural Language Processing Pacific Rim Symposium (NLPRS’01)*, Tokyo, Japan, pp. 533–538, 2001.

[SAN 07] SANJUAN E., IBEKWE-SANJUAN F., TORRES-MORENO J.-M., *et al.*, “Combining vector space model and multi word term extraction for semantic query expansion”, *12th International Conference on Applications of Natural Language to Information Systems (NLDB’07)*, vol. 4592 of LNCS, Paris, France, Springer-Verlag, pp. 252–263, 27–29 June 2007.

[SAN 12] SANJUAN E., MORICEAU V., TANNIER X., *et al.*, “Overview of the INEX 2012 tweet contextualization track”, *CLEF (Online Working Notes/ Labs/ Workshop)*, 2012.

[SAT 01] SATOSHI C.N., SATOSHI S., MURATA M., *et al.*, “Sentence extraction system assembling multiple evidence”, *2nd NTCIR Workshop*, Tokyo, Japan, pp. 319–324, 2001.

[SCH 94] SCHMID H., “Probabilistic part-of-speech tagging using decision trees”, *International Conference on New Methods in Language Processing*, Manchester, UK, pp. 44–49, 1994.

[SIL 00] SILBER H.G., MCCOY K.F., “Efficient text summarization using lexical chains”, *ACM Conference on Intelligent User Interfaces*, New York, NY, pp. 252–255, 2000.

[SJÖ 07] SJÖBERGH J., “Older versions of the ROUGEeval summarization evaluation system were easier to fool”, *Information Processing and Management*, vol. 43, no. 6, pp. 1500–1505, 2007.

[SKO 68] SKOROXOD’KO E.F., “Information retrieval system of the Ukrainian Institute of Cybernetics”, *International Congress on Scientific Information*, Moscow, 1968.

[SKO 71] SKOROXOD’KO E.F., “Adaptive method of automatic abstracting and indexing”, *IFIP Congress-711*, Booklet TA-6, Ljubljana, Yugoslavia, pp. 133–137, 1971.

[SPE 09] SPECIA L., CANCEDDA N., DYMETMAN M., *et al.*, "Estimating the sentence-level quality of machine translation systems", MARQUEZ L., SOMERS H., (eds.), *13th Conference of the European Association for Machine Translation (EAMT)*, Polytechnic University of Catalonia, Barcelona, Spain, 2009.

[SPÄ 72] SPÄRCK-JONES K., "A statistical interpretation of term specificity and its application in retrieval", *Journal of Documentation*, vol. 1, no. 28, pp. 11–21, 1972.

[SPÄ 93] SPÄRCK-JONES K., "What might be in a summary?", KNORZ G., KRAUSE J., WOMSER-HACKER C., (eds.), *Information Retrieval'93: Von der Modellierung zur Anwendung*, Universitatsverlag Konstanz, Constance, Germany, pp. 9–26, 1993.

[SPÄ 95] SPÄRCK-JONES K., ENDRES-NIGGEMEYER B., "Automatic summarizing", *Information Processing and Management*, vol. 31, no. 5, pp. 625–630, 1995.

[SPÄ 96] SPÄRCK-JONES K., GALLIERS J.R., *Evaluating Natural Language Processing Systems: An Analysis and Review*, LNCS, vol. 1083, Springer-Verlag, New York, 1996.

[SPÄ 97] SPÄRCK-JONES K., "Summarizing: where are we now? Where should we go", MANI I., MAYBURY M., (eds.), *ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Association for Computational Linguistics, Madrid, Spain, July 1997.

[STE 04] STEINBERGER J., JEZEK K., "Text summarization and singular value decomposition", YAKHNO T., ed., *3rd International Conference Advances in Information Systems*, LNCS, vol. 3261 Izmir, Turkey, Springer-Verlag, Berlin, pp. 245–254, 2004.

[SUN 05] SUN J.-T., YANG Q., LU Y., "Web-page summarization using clickthrough data", *28th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*, ACM, pp. 194–201, 2005.

[SVO 07] SVORE K., VANDERWENDE L., BURGES C.J., "Enhancing single-document summarization by combining RankNet and third-party sources", *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 448–457, 2007.

[SWA 00] SWAN R., ALLAN J., "Automatic generation of overview timelines", *23rd International Conference on Research and Development in Information Retrieval *(SIGIR '00)*, Athens, Greece, ACM, New York, pp. 49–56, 2000.

[TAB 06] TABOADA M., MANN W.C., "Applications of rhetorical structure theory", *Discourse Studies*, vol. 8, no. 4, pp. 567–588, August 2006.

[TAK 05] TAKAMURA H., INUI T., OKUMURA M., "Extracting semantic orientations of words using spin model", *43rd Meeting on Association for Computational Linguistics (ACL '05)*, Ann Arbor, MI, ACL, pp. 133–140, 25–30 June 2005.

[TEU 97] TEUFEL S., MOENS M., "Sentence extraction as a classification task", MANI I., MAYBURY M., (eds.), *ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 11 July 1997.

[TEU 99] TEUFEL S., MOENS M., "Discourse-level argumentation in scientific articles: human and automatic annotation", *ACL Workshop: Towards Standards and Tools for Discourse Tagging*, Maryland, , pp. 84–93, 1999.

[TEU 02] TEUFEL S., MOENS M., "Summarizing scientific articles: experiments with relevance and rhetorical status", *Computational Linguistics*, vol. 28, no. 4, pp. 409–445, 2002.

[TOR 02] TORRES-MORENO J.-M., VELÁZQUEZ-MORALES P., MEUNIER J.-G., "Condensés de textes par des méthodes numériques", *Journées d'Analyse Statistique de Documents (JADT '02)*, vol. 2, St Malo, France, pp. 723–734, 2002.

[TOR 07] TORRES-MORENO J.-M., Du textuel an numérique: analyse et classification automatique, thesis, LIA, Université d'Avignon et des Pays de Vaucluse, France, 2007.

[TOR 09] TORRES-MORENO J.-M., ST-ONGE P.-L., GAGNON M., *et al.,* "Automatic summarization system coupled with a question-answering system (QAAS)", *CoRR*, vol. abs/0905.2990, 2009.

[TOR 10] TORRES-MORENO J.-M., "Reagrupamiento en familias y lexematización automática independientes del idioma", *Inteligencia Artificial*, vol. 14, no. 47, pp. 38–53, 2010.

[TOR 10a] TORRES-MORENO J.-M., RAMIREZ J., "REG: un algorithme glouton appliqué au résumé automatique de texte", *Journées d'Analyse Statistique de Documents (JADT '10)*, Rome, Italy, 2010.

[TOR 10b] TORRES-MORENO J.-M., SAGGION H., DA CUNHA I., *et al.*, "Summary evaluation with and without references", *Polibits: Research Journal on Computer Science and Computer Engineering with Applications*, vol. 42, pp. 13–19, 2010.

[TOR 12] TORRES-MORENO J.-M., "Artex is another text summarizer", *CoRR*, vol. abs/1210.3312, 2012.

[TRA 08] TRATZ S., HOVY E., "Summarization evaluation using transformed basic elements", *Workshop Text Analysis Conference (TAC '08)*, Gaithersburg, MD, 2008.

[TUR 05] TURNER J., CHARNIAK E., "Supervised and unsupervised learning for sentence compression", *43rd Meeting on Association for Computational Linguistics (ACL '05)*, Ann Arbor, MI, ACL, pp. 290–297, 25–30 June 2005.

[TZO 01] TZOUKERMANN E., MURESAN S., KLAVANS J.L., "GIST-IT: summarizing email using linguistic knowledge and machine learning", *Workshop on Human Language Technology and Knowledge Management - Volume 2001 (HLTKM '01)*, Stroudsburg, PA, ACL, pp. 11:1–11:8, 2001.

[TZO 14] TZOURIDIS E., NASIR J., BREFELD U., "Learning to summarise related sentences", *The 25th International Conference on Computational Linguistics (COLING '14)*, Dublin, Ireland, ACL, 2014.

[VAN 79] VAN RIJSBERGEN C., *Information Retrieval*, 2nd ed., Butterworth-Heinemann, Newton, MA, 1979.

[VAN 07] VANDERWENDE L., SUZUKI H., BROCKETT C., *et al*, "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion", *Information Processing and Management*, Pergamon Press, Tarrytown, vol. 43, no. 6, pp. 1606–1618, 2007.

[VAS 63] VASILIEV A., "Automatic Abstracting and Indexing", *Automatic Documentation – Storage and Retrieval*, Moscow, Russia, United Nations Educational, Scientific and Cultural Association, pp. 1–12, 11–16 November 1963.

[VIV 01] VIVALDI J., RODRÍGUEZ H., "Improving Term Extraction by Combining Different Techniques", *Terminology*, vol. 7, no. 1, pp. 31-47, 2001.

[VIV 10a] VIVALDI J., DA CUNHA I., TORRES-MORENO J.-M., *et al*, "Automatic Summarization Using Terminological and Semantic Resources", *LREC'10*, Valletta, Malta, ELRA, 19–21 May 2010.

[VIV 10b] VIVALDI J., DA CUNHA I., TORRES-MORENO J.-M., *et al.*, "Generació automàtica de resums de textos especialitzats: experimentacions en llengua catalana", *Terminàlia*, vol. 1, pp. 26–32, 2010.

[WAN 04] WAN S., MCKEOWN K., "Generating Overview Summaries of Ongoing Email Thread Discussions", *International Conference on Computational Linguistics (COLING'04)*, Geneva, Switzerland, pp. 549–555, 23–27 August 2004.

[WAN 10] WAN X., LI H., XIAO J., "Cross-Language Document Summarization Based on Machine Translation Quality Prediction", *48th Meeting of the Association for Computational Linguistics (ACL'10)*, Uppsala, Sweden, ACL, pp. 917–926, 11–16 July 2010.

[WAS 08] WASZAK T., TORRES-MORENO J.-M., "Compression entropique de phrases contrôlée par un perceptron", *Journées internationales d'Analyse statistique des Données Textuelles (JADT'08)*, Lyon, France, pp. 1163-1173, 2008.

[WHI 01] WHITE M., KORELSKY T., CARDIE C., *et al.*, "Multidocument Summarization via Information Extraction", *1st International Conference on Human Language Technology Research*, HLT '01, Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 1–7, 2001.

[WIN 90] WINKLER, W. E., "String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage", *American Statistical Association*, pp. 354–359, 1990.

[WIT 99] WITBROCK M.J., MITTAL V.O., "Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries", *22nd Conference SIGIR'99*, Berkeley, CA, USA, ACM, pp. 315–316, 15–19 August 1999.

[YAT 07] YATSKO V., VISHNYAKOV T., "A Method for Evaluating Modern Systems of Automatic Text Summarization", *Automatic Documentation and Mathematical Linguistics*, vol. 41, no. 3, pp. 93–103, 2007.

[YIN 12] YIN W., PEI Y., ZHANG F., HUANG L., "SentTopic-MultiRank: a Novel Ranking Model for Multi-Document Summarization", *International Conference on Computational Linguistics (COLING'12)*, pp. 2977–2992, 2012.

[YOU 06] YOUSFI-MONOD M., PRINCE V., "Compression de phrases par élagage de l'arbre morpho-syntaxique", *Technique et Science Informatiques*, vol. 25, no. 4, pp. 437–468, 2006.

[YOU 07] YOUSFI-MONOD M., Compression automatique ou semi-automatique de textes par élagage des constituants effaçables: une approche interactive et indépendante des corpus, PhD Thesis, University of Montpellier II, France, 2007.

[ZHO 05a] ZHOU L., HOVY E., “Digesting Virtual “Geek”Culture: The Summarization of Technical Internet Relay Chats”, *43rd Meeting on Association for Computational Linguistics (ACL’05)*, Ann Arbor, MI, USA, ACL, pp. 298–305, 25–30 June 2005.

[ZHO 05b] ZHOU L., TICREA M., HOVY E.H., “Multi-document Biography Summarization”, *CoRR*, vol. abs/cs/0501078, 2005.

Index

Other titles from

iSTE

in

Cognitive Science and Knowledge Management

2014

DELPECH Estelle Maryline
Comparable Corpora and Computer-assisted Translation

MACHADO Carolina, DAVIM J. Paulo
Transfer and Management of Knowledge

SAAD Inès, ROSENTHAL-SABROUX Camille, GARGOURI Faiez
Information Systems for Knowledge Management

2013

LANDRAGIN Frédéric
Man-machine Dialogue: Design and Challenges

TURENNE Nicolas
Knowledge Needs and Information Extraction: Towards an Artificial Consciousness

ZARATÉ Pascale
Tools for Collaborative Decision-Making

2011

DAVID Amos
Competitive Intelligence and Decision Problems

LÉVY Pierre
The Semantic Sphere: Computation, Cognition and Information Economy

LIGOZAT Gérard
Qualitative Spatial and Temporal Reasoning

PELACHAUD Catherine
Emotion-oriented Systems

QUONIAM Luc
Competitive Intelligence 2.0: Organization, Innovation and Territory

2010

ALBALATE Amparo, MINKER Wolfgang
Semi-Supervised and Unsupervised Machine Learning: Novel Strategies

BROSSAUD Claire, REBER Bernard
Digital Cognitive Technologies

2009

BOUYSSOU Denis, DUBOIS Didier, PIRLOT Marc, PRADE Henri
Decision-making Process

MARCHAL Alain
From Speech Physiology to Linguistic Phonetics

PRALET Cédric, SCHIEX Thomas, VERFAILLIE Gérard
Sequential Decision-Making Problems / Representation and Solution

SZÜCS Andras, TAIT Alan, VIDAL Martine, BERNATH Ulrich
Distance and E-learning in Transition

2008

MARIANI Joseph
Spoken Language Processing