KTH VETENSKAP OCH KONST

KTH Computer Science
and Communication

Resource Lean and Portable Automatic Text Summarization

MARTIN HASSEL

Doctoral Thesis
Stockholm, Sweden 2007

TRITA-CSC-A 2007:9
ISSN-1653-5723
ISRN-KTH/CSC/A--07/09--SE
ISBN-978-91-7178-704-0

KTH School of Computer Science and Communication
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av filosofie doktorsexamen i datalogi måndagen den 11 juni 2007 klockan 13.15 i Sal E2, Lindstedtsvägen 3, KTH Campus Valhallavägen, Kungl Tekniska högskolan, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Today, with digitally stored information available in abundance, even for many minor languages, this information must by some means be filtered and extracted in order to avoid drowning in it. Automatic summarization is one such technique, where a computer summarizes a longer text to a shorter non-rendundant form. Apart from the major languages of the world there are a lot of languages for which large bodies of data aimed at language technology research to a high degree are lacking. There might also not be resources available to develop such bodies of data, since it is usually time consuming and requires substantial manual labor, hence being expensive. Nevertheless, there will still be a need for automatic text summarization for these languages in order to subdue this constantly increasing amount of electronically produced text.

This thesis thus sets the focus on automatic summarization of text and the evaluation of summaries using as few human resources as possible. The resources that are used should to as high extent as possible be already existing, not specifically aimed at summarization or evaluation of summaries and, preferably, created as part of natural literary processes. Moreover, the summarization systems should be able to be easily assembled using only a small set of basic language processing tools, again, not specifically aimed at summarization/evaluation. The summarization system should thus be near language independent as to be quickly ported between different natural languages.

The research put forth in this thesis mainly concerns three computerized systems, one for near language independent summarization – The HolSum summarizer; one for the collection of large-scale corpora – The KTH News Corpus; and one for summarization evaluation – The KTH eXtract Corpus. These three systems represent three different aspects of transferring the proposed summarization method to a new language.

One aspect is the actual summarization method and how it relates to the highly irregular nature of human language and to the difference in traits among language groups. This aspect is discussed in detail in Chapter 3. This chapter also presents the notion of "holistic summarization", an approach to self-evaluative summarization that weighs the fitness of the summary as a whole, by semantically comparing it to the text being summarized, before presenting it to the user. This approach is embodied as the text summarizer HolSum, which is presented in this chapter and evaluated in Paper 5.

A second aspect is the collection of large-scale corpora for languages where few or none such exist. This type of corpora is on the one hand needed for building the language model used by HolSum when comparing summaries on semantic grounds, on the other hand a large enough set of (written) language use is needed to guarantee the randomly selected subcorpus used for evaluation to be representative. This topic briefly touched upon in Chapter 4, and detailed in Paper 1.

The third aspect is, of course, the evaluation of the proposed summarization method on a new language. This aspect is investigated in Chapter 4. Evaluations of HolSum have been run on English as well as on Swedish, using both well established data and evaluation schemes (English) as well as with corpora gathered "in the wild" (Swedish). During the development of the latter corpora, which is discussed in Paper 4, evaluations of a traditional sentence ranking text summarizer, SweSum, have also been run. These can be found in Paper 2 and 3.

This thesis thus contributes a novel approach to highly portable automatic text summarization, coupled with methods for building the needed corpora, both for training and evaluation on the new language.

Sammanfattning

Idag, med ett överflöd av digitalt lagrad information även för många mindre språk, är det nära nog omöjligt att manuellt sålla och välja ut vilken information man ska ta till sig. Denna information måste istället filteras och extraheras för att man inte ska drunkna i den. En teknik för detta är automatisk textsammanfattning, där en dator sammanfattar en längre text till en kortare icke-redundant form. Vid sidan av de stora världsspråken finns det många små språk för vilka det saknas stora datamängder ämnade för språkteknologisk forskning. För dessa saknas det också ofta resurser för att bygga upp sådana datamängder då detta är tidskrävande och ofta dessutom kräver en ansenlig mängd manuellt arbete. Likväl behövs automatisk textsammanfattning för dessa språk för att tämja denna konstant ökande mängd elektronsikt producerad text.

Denna avhandling sätter således fokus på automatisk sammanfattning av text med så liten mänsklig insats som möjligt. De använda resurserna bör i så hög grad som möjligt redan existera, inte behöva vara skapade för automatisk textsammanfattning och helst även ha kommit till som en naturlig del av en litterär process. Vidare, sammanfattningssystemet bör utan större ansträngning kunna sättas samman med hjälp av ett mindre antal mycket grundläggande språkteknologiska verktyg, vilka inte heller de är specifikt ämnade för textsammanfattning. Textsammanfattaren bör således vara nära nog språkoberoende för att det med enkelhet kunna att flyttas mellan ett språk och ett annat.

Den forskning som läggs fram i denna avhandling berör i huvudsak tre datorsystem, ett för nära nog språkoberoende sammanfattning – HolSum; ett för insamlande av stora textmängder – KTH News Corpus; och ett för utvärdering av sammanfattning – KTH eXtract Corpus. Dessa tre system representerar tre olika aspekter av att föra över den framlagda sammanfattningsmetoden till ett nytt språk.

En aspekt är den faktiska sammanfattningsmetoden och hur den påverkas av mänskliga språks högst oregelbundna natur och de skillnader som uppvisas mellan olika språkgrupper. Denna aspekt diskuteras i detalj i kapitel tre. I detta kapitel presenteras också begreppet “holistisk sammanfattning”, en ansats tillsjälvutvärderande sammanfattning vilken gör en innehållslig bedömning av sammanfattningen som en helhet innan den presenteras för användaren. Denna ansats förkroppsligas i textsammanfattaren HolSum, som presenteras i detta kapitel samt utvärderas i artikel fem.

En andra aspekt är insamlandet av stora textmängder för språk där sådana saknas. Denna typ av datamängder behövs dels för att bygga den språkmodell som HolSum använder sig av när den gör innehållsliga jämförelser sammanfattningar emellan, dels behövs dessa för att ha en tillräckligt stor mängd text att kunna slumpmässigt extrahera en representativ delmängd lämpad för utvärdering ur. Denna aspekt berörs kortfattat i kapitel fyra och i mer önskvärd detalj i artikel ett.

Den tredje aspekten är, naturligtvis, utvärdering av den framlagda sammanfattningsmetoden på ett nytt språk. Denna aspekt ges en översikt i kapitel 4. Utvärderingar av HolSum har utförts både med väl etablerade datamängder och utvärderingsmetoder (för engelska) och med data- och utvärderingsmängder insamlade specifikt för detta ändamål (för svenska). Under sammanställningen av denna senare svenska datamängd, vilken beskrivs i artikel fyra, så utfördes även utvärderingar av en traditionell meningsextraherande textsammanfattare, SweSum. Dessa återfinns beskrivna i artikel två och tre.

Denna avhandling bidrar således med ett nydanande angreppssätt för nära nog språkoberoende textsammanfattning, uppbackad av metoder för sammansättning av erforderliga datamängder för såväl modellering av som utvärdering på ett nytt språk.

Acknowledgments

The stuff theses are made of does not thrive in a vacuum, on the contrary, it needs a prolificate environment and it would be ungracious of me to deny those deserving acknowledgment. In an incomplete and somewhat disorganized order, I lift my hat to:

- Hercules; did either of us believe that a simple question would lead to this? Your constant encouragement has been essential in seeing me through.
- Viggo; your sense for details has kept me aiming at the prize. The soffmöten has been an invaluable asset and a vital researchial critical mass.
- Jonas; for being a crucial part of that critical mass. Working with you is a pleasure.
- Ola; for allowing me to disturb you with music and questions, and for often having an answer. Simply put, roommate extraordinaire.
- Magnus; for also being part of that mythical critical mass.
- Magnus and Jussi; for showing me that random does not imply unstructured.
- ScandSum participants, most notably Koenraad and Anja; for fruitful discussions and inspiring retreats.
- Johan; for the stemmer and for most pleasurable dinner conversations.
- Knut and Alice Wallenberg's foundation; for giving me traveling grants, always almost getting me there.
- Kerstin; for always finding that extra little bit of money when traveling grants wouldn't quite get me there.
- NGSLT; for getting me to Nodalida and back again.
- Eva-Lena; for that shoulder I needed. Without your support I would not have stayed sane these years.
- Systemgruppen at KTH; for saving my behind in its time of utmost need.
- DaLi at SU, most notably Gunnel; without you I doubt this journey even would have started.
- Teachers and fellow students at GSLT; some of you I owe much of my LT knowledge to.
- Students who have taken our LT course; learning by teaching is a honor and a gift.
- Master students whom I've supervised; most of the time a blessing, you have taught me much.
- Henrik; for beer, music and impromptu conversation. All of them pleasing distractions.
- Basia; without you this thesis would not exist. It's been a rough ride coming this far, as per usual, with love.
- Lotta; for forcing me to take breaks when I needed them the most, and for still loving me when I didn't take them. I love you.
- Mother and father; for your undying trust, belief and support. Without you I would be nothing.
- Relatives; for acting safe haven and reference points in reality. For summers gone and come.
- Friends; for beer, laughs, games, film, music, comics, art, chili and so on we go.

Contents

Chapter 1

Introduction

Text summarization is the process of creating a summary of one or more texts. This summary may serve several purposes. One might, for example, want to get an overview of a document set in order to choose what documents one needs to read in full. Another plausible scenario would be getting the gist of a constant news flow, without having to wade through inherently redundant articles run by several news agencies, in order to find what might differ in reports from different parties. With digitally stored information available in abundance and in a myriad of forms, even for many minor languages, it has now become near impossible to manually search, sift and choose which information one should incorporate. Instead this information must by some means be filtered and extracted in order to avoid drowning in it. Automatic summarization is one such technique.

The title of this thesis sets the focus on summarization of text, automatically carried out by a computer program using as few human resources as possible. The resources that are used should to as high extent as possible be already existing, not specifically aimed at summarization and, preferably, created as part of natural literary processes. Moreover, the summarization system should be able to be easily assembled using only a small set of basic language processing tools, again, not specifically aimed at summarization. The summarization system should thus be near language independent as to be quickly ported between different natural languages.

The motivation for this is as simple as intuitive. Apart from the major languages of the world, there simply are a lot of languages for which large bodies of data aimed at language technology research, let alone research in automatic text summarization, are lacking. There might also not be resources available to develop such bodies of data, since it is usually time consuming and hence expensive. Nevertheless, there will still be a need for automatic text summarization for these languages in order to subdue this constantly increasing amount of electronically produced text.

1.1 Research Issues

As the title of this thesis indicates, I have investigated two main research topics in the context of automatic text summarization. These two desired properties are *resource lean* and *portable*. I will below try to give a short definition of these two concepts and how they are reflected in the research presented in this thesis. As evaluation is a natural part of any research these two properties are investigated both from the perspective of evaluation of summaries, as well as the actual automated summarization of text. Also, as given by the introduction to this chapter, this thesis concerns itself with texts written in some natural language.

Resource Lean

Resources can be many things, e.g. human resources, economic resources, spatial or temporal resources, data resources etc. The research on summarization evaluation presented in this thesis is in some sense lean on human resources since the *KTH News Corpus* tool-kit nearly automates the collection of large scale corpora suitable both for training and evaluation of summarization systems, and the *KTH eXtract Corpus* suite of tools guides human informants in the collection of gold[1] summary extracts used for evaluation, making the task almost effortless.

The research on language independent automatic text summarization, on the other hand, is certainly not lean on data resources. On the contrary, the *HolSum summarizer* demands large bodies of data for training. However, this data need not be annotated nor structured in any way, and can be collected “in the wild” (e.g. it can be text already in existence, produced for entirely different purposes). Thus it most certainly is resource lean regarding the other identified resource types, since:

1. structuring and annotation of data takes time and (most often) requires quite a bit of human effort;
2. human labor (usually) is more time consuming than the computerized counterpart;
3. humans desire more space than (most) computers require;
4. time, space and human labor most definitely cost money.

Portable

Also the portability of a system or method is an issue of cost. When lacking the necessary resources to build advanced systems the ability to transfer an intermediary system to other languages or domains, with as little effort as possible, becomes crucial. This applies both to the system itself as to the evaluation of the system in its new context.

[1] An “ideal” set of reference examples used for training or evaluation is in natural language processing often referred to as a gold-standard corpus, see Section 4.2.

The systems presented in this thesis are portable in more than one sense. The choice of programming languages have been those that are directly runnable on many platforms, e.g. Java, Perl, PHP, and Python. These are examples of programming languages that often are referred to as *platform independent*, platform here meaning computer environment. The choice of technologies for data representation has likewise fallen upon those regarded as movable between platforms (e.g. flat ASCII files and, especially, XML[2] documents). Also with respect to the natural languages the systems and methods presented herein should to a high extent be portable.

The *KTH eXtract Corpus* tool-kit, used for collection of gold-standard extracts for evaluation of summaries, is easily ported to other languages by simply translating the user interface (which is both minimal and skinned,[3] so the effort there is little). If the informants share a common (natural) language the effort is even less since they then can be presented with the same skin/language. The texts being summarized by the human informants should naturally be of the language one wishes to build a gold-standard extractive summary corpus for, and the informants should like-wise be fluent in this language, but the system itself does not pose any language restrictions.

The *HolSum summarizer* has so far only been evaluated on English and Swedish, but it does not make any requirements on the text it is processing bar it being segmentable into extractable text chunks ("sentences") and meaning bearing tokens ("words"). Even the plethora of code pages[4] is seamlessly handled by Java by its conversion between different code pages and its internal Unicode[5] definition.

Choice of Languages

As has already been mentioned the research presented in this thesis is, or should be, near language independent. However, the methods, tools and systems developed during this research have mainly been evaluated on Swedish and English, although some evaluation has been carried out on other languages by others than myself (Alonso Alemany and Fuentes Fort 2003, Dalianis *et al.* 2003, 2004, Hassel and Mazdak 2004, de Smedt *et al.* 2005). These evaluation runs are not reported herein, even though I have in many cases been involved in the transition between different languages. Swedish was chosen as one of the main evaluation languages based on the fact that at the onset of this research there existed no evaluation resources nor

[2] Extensible Markup Language is a general-purpose markup language that allows for the definition and representation of both binary data as well as almost any information in any written human language.

[3] A skinned user interface has its "appearance", in this case the instructions presented to the user in some natural language, stored separately from the actual source code. By not having the parts being localized obscured by code, the translation of these is dramatically simplified.

[4] A code page is a definition of how textual data is represented in flat text files and for most languages there exists one or more different definitions.

[5] Unicode is an industry standard definition designed to allow symbols ("letters") from all of the worlds writing systems to be consistently represented and manipulated by computers.

tools for evaluation of automatic text summarization for Swedish, and one of the few existing summarization systems, SweSum, had yet not been evaluated even for English let alone Swedish. Swedish was thus a prime candidate for the research motivated by this thesis.

The choice of verifying the research reported on in this thesis also on English is based on the fact that for English there already exists a framework for evaluation of automatic text summarization, with recurring evaluation campaigns where several summarization systems "compete" on the same data set. There is thus a possibility to compare the performance of a system not only to human performance, but also to put the system performance into the perspective of how the state-of-the-art systems geared at a specific language fare.

1.2 Main Contributions

This thesis contributes a framework for resource lean and highly portable automatic text summarization. This framework is threefold in providing methods for data collection and for evaluation, as well as for the core task, text summarization. In addition, an important contribution of this thesis is the notion of "holistic summarization"; a concept where both the text being summarized as well as the resulting summary are seen as a whole, and where the fitness of the summary is semantically assessed with respect to the original text. This concept is realized within the proposed framework.

1.3 Thesis Road Map

This thesis is organized into five chapters, of which you are currently reading the first. The following chapters will form the foundation on which this thesis rests. These chapters will be providing an introduction to automatic text summarization as a research field as well as detailing the methods and concepts relevant to the research carried out during the work with this thesis. At the end of each chapter a summary is given recapturing the topics most central to that chapter.

Chapter 2 presents an introduction to summarization and the role summaries play in different reading and writing activities. The concept of automatic text summarization, which is a term commonly used to denote summarization carried out by means of a computer program, is introduced. Followed by an overview of a selection of representative systems and approaches, we here also find a brief look into the what's, why's and when's of summarization.

Chapter 3, in turn, concerns the research carried out on language independent automatic text summarization, with a thorough background detailing several concepts and methodologies used and presented in the related papers (Hassel and Sjöbergh 2005, 2006, 2007), deliberating their language independence and how traits of different language families reflect on these methods. A topic most relevant to this research is that of how words come to gain their meaning through their use in natural language contexts. I conjunction to this discussion the use of word distribution patterns to model meaning is introduced.

Evaluation being a necessary factor in any research Chapter 4 puts the focus on evaluation of summaries and summarization. A distinction between intrinsic and extrinsic evaluation is made and a survey of how these two approaches have been applied to summarization evaluation is given. Here we also find an overview of tools and systems available for summarization evaluation, among these the KTH eXtract Corpus tool-kit (Hassel and Dalianis 2005). Also the collection and preparation of resources for summary evaluation is briefly discussed in the context of the KTH eXtract Corpus and the KTH News Corpus (Hassel 2001a, Dalianis and Hassel 2001).

The fifth and final chapter consists of a summary of the papers reprinted in this thesis, as well as a brief overview of the different corpora, tool packages and systems that have been collected, developed and applied during the course of the thesis research. At the end of this chapter we also find some concluding remarks followed by a brief glimpse into the future.

The thesis concludes with a reprint of a selected set of previously published papers that constitute the body of the research carried out during the work with this thesis.

1.4 Chapter Summary

In this chapter the motivation for the research presented in this thesis has been stated as developing *automatic text summarization* methods for languages lacking advanced *natural language processing* tools and large bodies of *annotated* or *structured* data. The summarization method should therefore effortlessly transfer from one language or domain to another. In this context the two main research topics *resource lean* and *portable* were presented and defined.

Since evaluation is a natural part of any research these two research topics were also briefly discussed in the light of *summary evaluation.* After a short presentation of the systems developed during the course of this thesis research, relating them to the two concepts resource lean and portable, a motivation for evaluating on *Swedish* as well as a major language as *English* was given. This motive was stated as the availability of *widely accepted data* and the comparability with state-of-the-art systems through *standardized evaluation schemes.*

Also given was a short declaration of the *main contributions* of the thesis, stated as defining a framework for *resource lean and highly portable automatic text summarization* together with the concept of *holistic summarization.* A *road map* to reading this thesis was also outlined, shortly *summarizing* chapter by chapter, thus giving an *overview* of how the thesis is organized.

Chapter 2

Summaries and the Process of Summarization

Automatic text summarization is the technique where a computer automatically creates an abstract, or summary, of one or more texts. The initial interest in automatic shortening of texts was spawned during the sixties in American research libraries. A large amount of scientific papers and books were to be digitally stored and made searchable. However, the storage capacity was very limited and full papers and books could not be fit into databases those days. Therefore summaries were stored, indexed and made searchable. Sometimes the papers or books already had summaries attached to them, but in cases were no ready-made summary was available one had to be created. Thus, the technique has been developed for many years (see Luhn 1958, Edmundson 1969, Salton 1988) and in recent years, with the increased use of the Internet, there have been an awakening interest for summarization techniques. Today the situation is quite the opposite from the situation in the sixties. Today storage is cheap and seemingly limitless. Digitally stored information is available in abundance and in a myriad of forms to an extent as to making it near impossible to manually search, sift and choose which information one should incorporate. This information must instead be filtered and extracted in order to avoiding drowning in it.

2.1 The World According to ISO

According to the documentation standard ISO 215:1986, a *summary* is a "brief restatement within the document (usually at the end) of its salient findings and conclusions, and is intended to complete the orientation of a reader who has studied the preceding text" while an *abstract* is, according to the same standard, a "Short representation of the content of a document without interpretation or criticism". An abstract as such is most often placed in the beginning of a text, for example in a scientific report. In this thesis, however, these two terms will be used somewhat interchangeably, as they commonly are in the research field of automatic text summarization. Instead it is common practise to differentiate between *abstraction* and *extraction*, both which are seen as processes of (automatic) summarization. With this terminology a distinct border is drawn between extraction-based, or cut-and-paste, summaries where the summary is composed of more or less edited fragments from the source text (this is the task of text extraction), as opposed to abstraction based summaries ("true abstracts") where the source text is transcribed into some formal representation and from this regenerated in a shorter more concise form (Hovy and Lin 1997). A good overview of the field can be found in Mani and Maybury (1999).

2.2 In Defense of the Abstract

Why do we need automatic text summarization, indeed, why do we need summaries or abstracts at all? In the words of the American National Standards Institute (ANSI) – "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". Actually the abstract is highly beneficial in several information acquisition tasks, some examples are given in (Borko and Bernier 1975):

- Abstracts promote current awareness
- Abstracts save reading time
- Abstracts facilitate selection
- Abstracts facilitate literature searches
- Abstracts improve indexing efficiency
- Abstracts aid in the preparation of reviews

Furthermore, human language is highly redundant, probably to facilitate error recovery in highly noisy channels. Mathematician and electrical engineer Claude E. Shannon has, for example, using a training data of 583 million words to create a trigram language model and a corpus of 1 million words for testing, shown a 75% redundancy of English on letter level (Shannon 1948). Shannon initially defined redundancy as "the discovery of long-windedness" and accordingly it is not the amount of information that is increased, but the probability that the information reaches the recipient.

Fittingly, entropy experiments have also shown that humans are just as good at guessing the next letter – thus discerning the content of the text on a semantic level – after seeing only 32 letters as after 10,000 letters (Burton and Licklider 1955). Other experiments (Morris *et al.* 1992) concerning reading comprehension of extraction based summaries compared to full documents have shown that extracts containing 20% or 30% of the source document are effective surrogates of the source document. Performance on 20% and 30% extracts is no different than informative abstracts.

Then, how does one go about constructing an abstract? Cremmins (1996) gives us the following guidelines from the American National Standard for Writing Abstracts:

- State the purpose, methods, results, and conclusions presented in the original document, either in that order or with an initial emphasis on results and conclusions.
- Make the abstract as informative as the nature of the document will permit, so that readers may decide, quickly and accurately, whether they need to read the entire document.
- Avoid including background information or citing the work of others in the abstract, unless the study is a replication or evaluation of their work.
- Do not include information in the abstract that is not contained in the textual material being abstracted.
- Verify that all quantitative and qualitative information used in the abstract agrees with the information contained in the full text of the document.
- Use standard English and precise technical terms, and follow conventional grammar and punctuation rules.
- Give expanded versions of lesser known abbreviations and acronyms, and verbalize symbols that may be unfamiliar to readers of the abstract.
- Omit needless words, phrases, and sentences.

In automatic abstracting or summarization, however, one often distinguishes between informative and indicative summaries, where informative summaries intend to make reading of source unnecessary, if possible. Indicative summaries, on the other hand, act as an appetizer giving an indication of the content of the source text, thus making it easier for the reader to decide whether to read the whole text or not.

2.3 Automatic Text Summarization

Automatic summarization is the creation of a briefer representation of a body of information by a computer program. The product of this procedure should still contain the most central facts of the original information. Automatic text summarization, thus analogously, is the shortening of texts by computer, while still retaining the most important points of the original text.

Automatic text summarization is a multi-faceted endeavor indeed. Churning under the threat of information overload[1] the research field has branched out in several dimensions. There is no clear-cut path to follow in classification, and summarization systems usually tend to fall into several categories at once. If we first broadly define three aspects of a summarizing system as i) source, representing the multitude of input formats and possible origins of the information being summarized, ii) purpose,

[1] The term "information overload" commonly refers to the state of having too much information to remain informed, and thereby rendered unable to make a decision, about a particular topic.

being the intended use for the generated summary, and iii) composition, denoting the output format of the summary and the information contained therein; we can then, according to (Spärck-Jones 1999, Lin and Hovy 2000, Baldwin *et al.* 2000) among others, roughly make the following, by necessity inconclusive, division:

Source (Input):

- Source: single-document vs. multi-document
- Language: mono-lingual vs. multi-lingual vs. cross-lingual
- Genre: news vs. technical report vs. scientific paper etc.
- Specificity: domain-specific vs. general
- Length: short (1–2 pages) vs. long (> 50 pages)
- Media: text, graphics, audio, video, multi-media etc.

Purpose:

- Use: generic vs. query-oriented (aimed to a specific information need)
- Purpose: what the summary is used for (e.g. alert, preview, inform, digest, provide biographical information)
- Audience: untargeted vs. targeted (aimed at a specific audience)

Composition (Output):

- Derivation: extract vs. abstract
- Format: running text, tables, geographical displays, time lines, charts, illustrations etc.
- Partiality: neutral vs. evaluative (adding sentiment/values)

The generated summaries can also be divided into different genres depending on their intended use, for example: headlines, outlines, briefings, minutes, biographies, abridgments, sound bites, movie summaries, chronologies etc. (Mani and Maybury 1999). Consequently, a summarization system falls into at least one, often more than one, slot in each of the main categories above, and thus must also be evaluated along several dimensions using different measures.

Approaches to Automatic Text Summarization

Summarization approaches are often, as mentioned, divided into two main groups, text extraction and text abstraction. Text abstraction, being the more challenging task, is to parse the original text in a deep linguistic way, interpret the text semantically into a formal representation, find new more concise concepts to describe the text and then generate a new shorter text, an abstract, with the same basic information content. This is in many aspects similar to what humans abstractors do when writing an abstract, even though professional abstractors also do utilize surface-level information such as headings, key phrases and position in the text as well as the overall organization of the text into more or less genre specific sections (Liddy 1991, Endres-Niggemeyer *et al.* 1995, Cremmins 1996). The parsing and interpretation of text is a venerable research area that has been investigated for many years. In this area we have a wide spectrum of techniques and methods ranging from word by word parsing to rhetorical discourse parsing as well as more statistical methods, or a mixture of all. Also the generation of text is a vigorous research field with techniques ranging from canned text and template filling to more advanced systems with discourse planners and surface realizers.

Text extraction, on the other hand, means to identify the most relevant passages in one or more documents, often using standard statistically based information retrieval techniques augmented with more or less shallow natural language processing and genre or language specific heuristics. These passages, often sentences or phrases, are then extracted and pasted together to form a non-redundant summary that is shorter than the original document with as little information loss as possible. Sometimes the extracted fragments are post-edited, for example by deleting subordinate clauses or joining incomplete clauses to form complete clauses (Jing and McKeown 2000, Jing 2000).

Language Independent Approaches

Gong and Liu (2001) use Latent Semantic Analysis (see Section 3.3) in a generic text summarization system that creates text summaries by ranking and extracting sentences from the original document. This system employs LSA to create "synonym sets", or rather *semantic sets*, which are used to pin point the topically central sentences. Each set is only used once so that only sentences that are highly ranked and different from each other are selected. This in an attempt to create a summary with a wider coverage of the document's main content while reducing the redundancy of the summary.

TextRank presents an interesting approach to unsupervised extractive summarization that employs iterative graph-based ranking algorithms to encode the cohesive structure of a text (Mihalcea 2004, 2005). The system claims no need of language-specific knowledge resources or manually constructed training data, which should make it highly portable to new languages or domains. However, as with the approach detailed in this thesis, the system still requires language specific knowledge in the form of segmentation rules (see Section 3.1), as well as a stop list (see Section 3.1) if the suggested stopword filter is to be used

Another approach, by Yeh *et al.* (2002), also makes use of LSA, this time in order to derive a semantic matrix of a document. Based on this, semantic sentence representations are used to construct a text relationship map (Salton *et al.* 1997) for interpreting conceptual structures of a document. The significance of a sentence, with respect to the source document, is then measured by counting the number of links that it has.

Application Areas

The application areas for automatic text summarization are extensive. As the amount of information on the Internet grows abundantly, it is difficult to select relevant information. Information is published simultaneously on many media channels in different versions, for instance, a paper newspaper, web newspaper, SMS[2] message, mobile radio newscast, and a spoken newspaper for the visually impaired. Customization of information for different channels and formats is an immense editing job that notably involves shortening of original texts.

Automatic text summarization can automate this work completely or at least assist in the process by producing a draft summary. Also, documents can be made accessible in other languages by first summarizing them before translation, which in many cases would be sufficient to establish the relevance of a foreign language document, and hence save human translators work since they need not translate every document manually. Automatic text summarization can also be used to summarize a text before an automatic speech synthesizer reads it, thus reducing the time needed to absorb the key facts in a document. In particular, automatic text summarization can be used to prepare information for use in small mobile devices, such as a PDA,[3] which may need considerable reduction of content.

2.4 Chapter Summary

In this chapter we have taken a look at common definitions of what constitutes a *summary* respectively an *abstract*. An abstract is defined as a short representation of the contents, usually located at the top of a document. A summary, on the other hand, is a short concluding restatement of the documents most central findings located at the very end of the document. We have also defined how these two terms are applied in the field of automatic text summarization, distinguishing between *abstraction* and *extraction*. A statement of the *serviceability* of abstracts in daily reading tasks was also given.

Following a contrastive description of how the task of *manual summarization* ought to be carried out, a layout of different aspects of *automatic summarization* was presented. Some related near *language independent* approaches have also been discussed. Finally, we took a brief look at some tasks and contexts that might benefit from automatic text summarization.

[2] *Short Message Service*, the transmission of short text messages to and from a mobile phone, fax machine and/or IP address. Messages must be no longer than 160 alpha-numeric characters.

[3] *Personal Digital Assistant* small mobile hand-held device that provides computing and information storage and retrieval capabilities, often contains calendar and address book functionality.

Chapter 3

Language Independent Summarization

A distinction most pertinent to this thesis is that of *language dependent* and *language independent* natural language processing (NLP). A language dependent system would be a system geared at a specific language, or a set of languages. It might perhaps utilize manually built lexical resources such as ontologies, thesauri or other language or domain specific knowledge bases. Other dependencies constraining a system to a specific language may be the employment of advanced tools as, for example, full parsers, semantic role assigners or named entity tagging, or the use of techniques such as template filling. The term "language independent", on the other hand, usually denotes a NLP system that is easily transferred between languages or domains. The system is thereby independent of the target language.

Also, it is important to distinguish between *language independence* and *cross-language*, where the latter means that the system processes two more more languages during the same task. One example of this would be cross-language information retrieval (CLIR), wherein a system retrieves information written in a language different from the language of the user's search query. Also in automatic text summarization there is an interest in cross-language summarization, e.g. the summarization of documents written in one language, with the summary presented in another (Lenci *et al.* 2002, de Chalendar *et al.* 2005, Siddharthan and McKeown 2005, Pingali *et al.* 2007). Another example is the summarization of documents where various parts of the text are written in different languages. The latter might be the case if a text mainly written in one language contains quotations in another, which then appear in translated form in the summary (if included).

The term "cross-language" has many synonyms; cross-lingual, multi-lingual and trans-lingual perhaps being the most frequent. However, "multi-lingual summarization" sometimes is used to denote a summarization system that covers several languages, but where the input and output language stays the same throughout the task. This does not necessarily imply language independence, no more than "cross-language" does, since this coverage might just as well be hard-coded for each language.

A *true* language independent NLP system should be directly transferable to new domains or a completely different natural language. As will be discussed in this thesis, there are many steps to cover on the way to a portable summarization system. No matter to what extent an extractive summarization system is, or claims to be, language independent it still has to do some more or less language dependent preprocessing. At the very least some knowledge about natural languages in general must be made available to the system in order to facilitate segmentation of the text into desirable units of extraction.

3.1 Preprocessing

Prior to any deeper linguistic treatment of a text the units of that text must be demarcated and possibly classified. The text can initially be viewed as a mere sequence of characters within which we must define these units (Grefenstette and Tapanainen 1994). First after having defined and isolated the units we are interested in we can begin to operate on them. This stage of isolation occurs on many levels; e.g. tokenization divides the character sequence into words, sentence splitting further divides sequences of words into sentences, and so on. Besides dividing the text into desired units, other forms of preprocessing might also be done before the main task is carried out. Examples of such preprocessing steps are forming common representations for different forms of the same word (stemming or lemmatization) and removing words uninteresting, or even harmful, for latter linguistic processing (stopword filtering). These preprocessing steps are often more or less language dependent, and thus deserve a brief discussion.

Segmentation

In order to perform extractive summarization we must first decide on what granularity our extraction segments will have, i.e. the "size" of the textual units that we copy from the original text and paste into our summary. This could be a paragraph, a sentence, a phrase or even a clause, although the most common probably is extraction performed on sentence level.

Often it is necessary to first split the text into words (tokens) in order to correctly identify these boundaries between clauses, phrases or sentences. Sentence splitting as such is often considered as a non-trivial task, considering the irregularities of natural languages. However, at least for many Germanic and Romance languages a small set of regular expressions,[1] perhaps accompanied by a list of abbreviations commonly including a punctuation mark, usually produces an

[1] A regular expression is a string that, internally, is used to describe or match a set of strings, according to a given syntax.

acceptable result. Using a list of abbreviations of course makes the tokenization inherently language dependent as an abbreviation list usually is hand-crafted.

Apart from being a preprocessing stage to splitting the text into extraction segments (e.g. sentences), tokenization is also necessary in order to perform various statistical operations. Again, tokenization for many Germanic and Romance languages can often satisfyingly be accomplished with a small set of regular expressions defining a word as a token separated by white-space and/or a punctuation mark.[2] There are, though, languages lacking word-boundary markers, such as Chinese and Japanese, which certainly provide a more challenging task (Crystal 1987), but much statistical work has been carried out also for these languages, e.g. Chinese word segmentation (Luk 1994).

Stemming

In running text the same word usually occurs in several different morphological variants. These inflected forms are governed by the context, i.e. if it is presented in singular or plural form, present or past tense etc. In most cases these different lexical forms have similar semantic interpretations and can consequently often be considered as equivalent for the purpose of many Information Management applications. In order for an information management system to be able to treat these inflected forms as one concept, often referred to as a lexeme, it is common to use a so-called stemming algorithm, or simply, a stemmer.

These stemmers attempt to reduce a word to a common root form, often called a stem, so that the words in a document can be represented by one lexical string (term) rather than by the original word forms. The effect is not only that different variants of a term can be conflated to a single representative form, but it also reduces the size of the vocabulary the system need to store representations for. This means that the number of distinct terms needed for representing a document, or a set of documents, usually is dramatically reduced. In many cases this smaller dictionary size results in a saving of storage space and processing time, as well as making document representations (see Section 3.2) less noisy, more dense and more versatile.

From an information management perspective it usually does not matter whether the generated stems are genuine words or not. The internal representation of *computation*, *computations*, *computational* and *computer* might all be stemmed to the same root *comput*. However, the efficiency of the stemmer depends on whether different words with the same "base meaning" are conflated to the same root, and words with distinct meanings are kept separate. Related to this problem are the two terms overstemming and understemming.

Overstemming occurs when two words are given the same stem where they in fact should not be. For instance, if *experiment* as well as *experience* are transformed

[2] Considered punctuation marks could be full stop, comma, question and exclamation marks etc.

to *experi* this would cause the meanings of the stems to be diluted, which in turn will decrease precision of the language model. This can be viewed as taking off too large a suffix.

Understemming, on the other hand, occurs when words that should be brought to the same base form are not. One example would be *running* being transformed to *run*, and *ran*, instead of being also transformed to *run*, being "transformed" to *ran*. Another example would be removing a too short suffix, as in the case of reducing *adheres* to *adher*, and *adhesion* to *adhes*, respectively. Both these cases causes compound information to be spread over various stems, in our case *run/ran* and *adher/adhes*. This affects overall recall negatively in the manner that we cannot match a document written in present tense, about running, with a document written in past tense on the same topic.

An algorithm which instead attempts to transform an inflected word to its linguistically correct root, or lemma, is often called a lemmatizer. An advantage of using stems, rather than lemmas, in information management is that words belonging to different inflectional paradigms[3] but denoting semantically related concepts can be grouped together. For example, the words "calculate" (verb) and "calculation" (noun) could under most circumstances be said to denote very similar semantic concepts, and could thus be conflated to the common stem "calcul" in order to cover both when encoding the semantic content of various documents. It is quite obvious that, if correctly implemented, a stemmer further aids the counting of word occurrences by assigning a consolidated frequency for morphological variants and other semantically related words (see Section 3.2).

Isolating and Synthetic Languages

The level of morphological complexity strongly affects the complexity of the stemming task. In synthetic languages words are mainly composed of a root morpheme and a set of attached "bound" morphemes, carrying different syntactic meanings. For example, since Slovene is morphologically more diverse than English, a Slovene stemmer described in Popovic and Willett (1992) removes around 5 200 different suffixes. As a contrast, the English Porter Stemmer removes about 60 suffixes (Porter 1980).

On the other hand, in isolating languages the vast majority of the morphemes are free morphemes (Crystal 1987). As such they are considered to be full-fledged "words" and stemming can basically be seen as a subtask to the removal of stopwords, see Section 3.1.

[3] Words belonging to different word classes follow different inflectional paradigms, i.e. they are inflected differently even if their lemma, or one or more inflected forms by chance happen to coincide.

Non-Concatenative Morphology

Semitic languages, like Arabic and Hebrew, are considered challenging languages to perform stemming on, due to the omission of vowels in written language and an extensive use of non-concatenative morphology (McCarthy 1981). Arabic, for example, besides being a highly inflected language, has a form of non-concatenative morphology called transfixation, where different vowels inserted between the consonants give different semantically-related meanings to the resulting word (Larkey *et al.* 2002). Other forms of non-concatenative morphology, appearing in some languages, are the reduplication of parts of the word or the whole word itself, and deletion where a part of the word is simply removed.

All of these cases, of course, affect the level of difficulty in constructing a stemmer for different languages. Swedish, for example, only uses non-concatenative morphology in a few cases; such as “springa” (run), “sprang” (ran) and “sprungit” (ran). These are, however, so few that they are easily handled by a set of exception rules. English exhibits even fewer cases, e.g. “foot” which becomes “feet” in plural.

The Case of Language Independence

Efforts towards statistical language independent stemming have been taken, so this step can possibly be automated in a language independent system. A promising such approach, where stem classes are built using co-occurrence statistics, has been proposed by Xu and Croft (1998). They have demonstrated an improvement in information retrieval after clustering stem classes for English and Spanish, but for more morphologically complex languages the challenge of language independence still awaits.

Another such approach by Bacchin *et al.* (2002) treats prefixes and suffixes as a community of sub-strings. They attempt to discover these communities by means of searching for the best word splits, which in turn give the best word stems.

Compound Splitting

Most statistical language models are more or less susceptible to sparse-data issues. In reality this means that the presence of very rare words, or patterns, adds noise to the model since the statistical grounds for modelling a representation of these are too weak. One such phenomenon is compounding. A compound word is a word that consists of more than one lexeme. Agglutinative languages, which are languages in which most words are formed by joining morphemes together, tend to be very productive in creating compound words. Theoretically the length of a compound word is unlimited, however longer compounds do become unwieldy and are infrequent in actual discourse.

For example, the somewhat jocular Swedish compound noun

```
barnvagnshjulsekeruträtarlärling
```

would translate into English, being a mostly analytic language,[4] as "perambulator wheel spoke straightener apprentice". This 32-letter word of course makes for a very rare occurrence even in a gigaword corpus. It should thus be perfectly clear that the representation of a document's content would benefit highly from having each constituting lexeme represented separately, thereby consolidating frequencies. Consequently, related to the task of stemming is that of compound splitting.

In the Swedish example given above some of the lexeme boundaries are marked while others are not. If we split the compound into separate lexemes;

```
barn-vagn-s-hjul-s-eker-uträtar-lärling
```

we can find two instances of a genitive case marker, `-s`, that hint of a lexeme boundary, in the rest of the cases in this example there is no such hint. Also, in some cases the "hint" may be obfuscated, as in the case of `glasskål`, which may be separated into `glas-skål` (glass bowl), `glass-skål`[5] (ice cream bowl) and even `glass-kål` (ice cream cabbage). Several statistical approaches to identifying lexeme boundaries in compounds exist, and may be used in different combinations (Sjöbergh and Kann 2004).

Compound words come in several flavors. The two forms of compounds that are most beneficial to split, from a language modelling point of view, are endocentric and incorporative compounds. Endocentric compounds have a head lexeme which defines the base meaning and one or more modifying lexemes restricting this meaning, e.g. "steamboat". An example of composition by incorporation is when a noun is strung onto a verbal head, as in "breastfeed". In both these cases the compounds usually can be split and the lexemes be represented individually in a document description without risking to "distort" the topic of the text.

Exocentric compounds, on the other hand, may instead add noise to the model when split since their meaning often cannot be deduced from what their constituent parts separately denote. Examples of exocentric compounds are "manhandle" and "white-collar", where it would make little sense to split these words into separate lexemes. Rather, it would in fact lead to improper semantic associations.

Stopword Filtering

Stopword filtering is a common technique used to counter the obvious fact that many of the words contained in a document do not contribute particularly to the description of the documents content. For instance, words like "the", "is" and "and" contribute very little to this description, and in many cases they do in fact instead add noise.

[4] In analytic languages the syntax and meaning are shaped more by using particles and word order than by inflection. The concept is related to that of isolating languages (Crystal 1987).

[5] The letter trigram `sss` is not an allowed consonant cluster in the Swedish language, wherefore the `ss-s` sequence is reduced to `ss` in the joint between the lexemes.

This concept of noise reduces the usefulness and informative quality of representations of a document's content due to the fact that the often very frequent function words "drown" the less frequent content words, that better describe the document. Therefore it is common to remove these so-called stopwords prior to the construction of these document descriptions, leaving only the content bearing words in the text during processing. This removal of stopwords can be done on several different grounds.

Often a predefined, hand-crafted stop list containing common words is used for removal of words known to be function words, which although far more high frequent than content words are far fewer in number. This does, however, make the creation of document content representations language dependent.

Instead, one could, for example, apply term frequency thresholds, where both terms (words) that have a very high and those that have a very low frequency are removed, thereby reducing the noise. However, frequency thresholding requires two processing phases. First one has to go through a large collection of documents, a corpus, in order to build a statistically reliable frequency list containing all words found in the corpus. Then, based on this frequency list, one filters out the words that are less or more frequent than the lower and upper thresholds, respectively.

3.2 Document Signatures

The approach to language independent summarization presented in this thesis heavily relies on the notion that documents, or rather a source document and a set of proposed summaries, can be compared for similarity.

This notion is well established in, for example, Information Retrieval, where user queries act as fuzzy "descriptions" that are matched to a set of documents in order to find the document most similar to that description. When comparing documents for content similarity it is common practice to produce some form of document signatures. These signatures represent the content in some way, often as a vector of features, which are used as basis for such comparison. This comparison can be attempted on several levels, e.g. on lexical, syntactic or semantic grounds.

The Vector Space Model

The Vector Space model is a document similarity model commonly used in Information Retrieval (Salton 1971). In this model the document signatures are represented as feature vectors consisting of the words that occur within the documents, with weights attached to each word denoting its importance for the document (see Section 3.2). We can, for example, for each term (word) record the number of times it occurs in each document. This gives us what is commonly called a document-by-term matrix, $M_{d,t}$ below, where the rows represent the documents in the document collection and the columns each represent a specific term existing in any of the documents (a weight can thus be zero).

$$M_{d,t} = \begin{pmatrix} w_{1,1}, & w_{1,2}, & w_{1,3}, & \dots, & w_{1,n} \\ w_{2,1}, & w_{2,2}, & w_{2,3}, & \dots, & w_{2,n} \\ w_{3,1}, & w_{3,2}, & w_{3,3}, & \dots, & w_{3,n} \\ \dots, & \dots, & \dots, & \dots, & \dots \\ w_{m,1}, & w_{m,2}, & w_{m,3}, & \dots, & w_{m,n} \end{pmatrix}$$

If we, using this matrix, view the feature vectors as projections in a multi-dimensional space, where the dimensionality is given by the m number of documents and n number of index terms, we can then measure the lexical similarity between two documents simply by calculating the angle between these vectors in space. For example, if we want to measure the similarity between two documents, we can represent these with the following two vectors:

$$\begin{aligned} \overrightarrow{d_i} &= (w_{i,1}, w_{i,2}, w_{i,3}, \dots, w_{i,n}) \\ \overrightarrow{d_j} &= (w_{j,1}, w_{j,2}, w_{j,3}, \dots, w_{j,n}) \end{aligned}$$

In this notation $\overrightarrow{d_i}$ and $\overrightarrow{d_j}$ denote the two documents, with n being the total number of index terms occurring in these documents. We can now compute the similarity by calculating the cosine angle between the two:

$$cos(\overrightarrow{d_i}, \overrightarrow{d_j}) = \frac{\sum_{k=1}^{n}(w_{k,i} \times w_{k,j})}{\sqrt{\sum_{k=1}^{n}(w_{k,i})^2 \times \sum_{k=1}^{n}(w_{k,j})^2}}$$

Here n is the total number of terms recognized by the matching system, while $w_{k,i}$ and $w_{k,j}$ represent the importance of the index term k to $\overrightarrow{d_i}$ and $\overrightarrow{d_j}$, respectively. For our purposes, the two documents being compared for similarity might as well be a document being summarized coupled with a summary of said document, as we shall see in Section 3.4.

Term Weighting

When constructing document signatures it also common to modify word frequency counts in the hope of promoting semantically salient words. There are many theories on how to model salience, where the most common probably is the *tf·idf* model. In this model *tf* represents the term frequency, and corresponds to the number of times a certain content word, represented by its stem/lemma, occurs within a specific document. Often the number of terms is divided by the total number of terms in a given document, as to not unduly promote long documents when comparing documents, giving us:

$$tf_i = \frac{n_i}{\sum_k n_k}$$

However, much as with the much more frequent function words adding noise when attempting to identify content words describing the content of a document, in the same manner can very common content words "drown" content words describing a specific document, e.g. within a specific domain. One example of this would be that if we have set of documents discussing the medical treatment of cancer, then certain domain specific content words would to a high extent exist with a high frequency in each of the documents. In order to counter this phenomenon of domain specificity it is common weigh the term frequency by in how many documents the term occurs (Spärck-Jones 1972). This notion of document specificity is often referred to as the inverse document frequency, or simply *idf*. The final weight denoting a terms topical importance to a specific document in a set of documents thus can be defined as:

$$tfidf_i = tf_i \times \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

or simply $tf{\cdot}idf$.

As defining salience as frequency fluctuations between documents, the $tf{\cdot}idf$ model requires a set of documents as well as the necessity of examining all of them noting in which documents a specific term occurs in order to calculate the weight of each term.

To overcome this several other approaches to capturing salience have been put forth. One such, proposed by Ortuño *et al.* (2002), models salience by tracking the distributional pattern of terms within a document. They show that the spatial information of a word is reliable in predicting the relevance of that word to the text being processed, independently of its relative frequency. The base of this observation is that words relevant to a text will normally appear in a very specific context, concentrated in a region of the text, presenting large frequency fluctuations, i.e. keywords come in bursts.

The burstiness of a word is here calculated using the standard deviation of the distance between different occurrences of the same word in the text. Though, words that occur only with large distances between occurrences usually have a high standard deviation by chance, so the standard deviation is divided by the mean distance between occurrences. With μ being the mean and σ the standard deviation of the distances between occurrences, in words, the final relevance weight of a term k thus is:

$$term_k = \frac{\sqrt{s^2 - \overline{s}^2}}{\mu}$$

However, the problem with counting terms on a lexical level is that the relation between the terms is not always what they seem to be, at least not by only looking at the constituting characters. Rather, the relation between words and concepts is many-to-many. For example, we have synonymy, where a number of words with same "meaning" have very different lexical appearances. In this case lexical term matching misses relevant frequency conflations thus impacting recall negatively. A hypothetical example would that we have document D such as $D = \{kitten, dog, pussy, cat, mouser, doggie, feline\}$. It is quite obvious, given that you know the meanings of the words occurring in the document, that the document is about cats. However, the traditional $tf{\cdot}idf$ model would not necessarily recognize this fact.

3.3 The Meaning of Words

Modeling of the meaning of words has always been an elusive task in natural language processing (Sahlgren 2002). Words are nothing more than sounds or a sequence of graphemes[6] until they become associated with an object, an action or some characteristic. Words therefore not only come to denote objects, phenomena or ideas in the physical world, but also gain a connotative substance based on how and when they are used (Mill 1843). In the Saussurean tradition this connotation, or meaning, is seen as to arise from the relative difference between words in a linguistic system. According to Saussure this constantly restructured system of differences is negotiated through social activity in a community of users (Saussure 1916). Two types of relations constitute the base of this difference, where syntagmatic relations concern positioning and paradigmatic relations act as functional contrasts (substitution). The meaning of a word is thus defined by the company it keeps, or does not keep.

Semantic Vector Space Models

Until the early nineties most of the work in statistical language learning was concentrated on syntax (Charniak 1993). However, with the induction of Latent

[6] Graphemes are the smallest inseparable units (the "atoms") of a writing system and include letters, numerals and punctuation marks, as well as logographs in e.g. Chinese and syllables in syllabric languages like Japanese.

Semantic Analysis (Dumais *et al.* 1988) a whole new field of lexical statistical semantics sprang into existence and today enjoys considerable attention in current research on computational semantics.

The general idea behind semantic vector space models, or word space models, is to use statistics on word distributions in order to generate a high-dimensional vector space. In this vector space the words are represented by context vectors whose relative directions are assumed to indicate semantic similarity. The basis of this assumption is the *distributional hypothesis* (Harris 1954), according to which words that occur in similar contexts also tend to have similar properties (meanings/functions). From this follows that if we repeatedly observe two words in the same (or very similar) contexts, then it is not too far fetched to assume that they also mean similar things, or at the very least to some extent share properties. Furthermore, depending on how we model these contexts we should be able to capture different traits. We should for instance be able to populate the word space model with syntagmatic relations if we collect information about which words that tend to co-occur, and with paradigmatic relations if we collect information about which words that tend to share neighbors (Sahlgren 2006).

This approach is often seen as a solution to semantic difficulties in NLP like synonymy and hyperonymy, which are not captured by the traditional vector space model (see Section 3.2). It should however be noted that semantic vector space methods still are oblivious to the nature of the lexical strings they are tracking. Therefore, given a definition of what constitutes a meaning bearing token in a particular language, it usually is advantageous to perform stemming and/or stopword removal, depending on e.g. the level of morphology of the language in question. Such considerations taken into account a word space model can be applied to basically any language. For instance, word space models have been applied to, among many other languages; Swedish, English, German, Spanish as well as Japanese (Sahlgren *et al.* 2002, Hassel 2005, Sahlgren and Karlgren 2005, Grönqvist 2006).

Latent Semantic Analysis

One of the earliest and still most popular word space model is Latent Semantic Analysis (Deerwester *et al.* 1990, Landauer *et al.* 1998). Latent Semantic Analysis (LSA) builds upon in the assumption that there exists a latent structure in word usage that is obscured by the variability in word choice. The semantic content ("the message") can be viewed as a signal where the use of e.g synonyms constitute additive noise.

In LSA, the latent semantic vector space is calculated using co-occurrence statistics collected from a large set of documents. As with the traditional vector space model a document-by-term matrix is built, resulting in an extremely high-dimensional vector space. In this space each word represents one dimension, with all dimensions being orthogonal to each other. A cumbersome property of this representation is that it contains one dimension for each lexical token (word), even if some lexically dissimilar tokens may well have very similar meanings.

What is done in the case of LSA is to create a projection from the extremely high number of dimensions in the original matrix to a much lower number of dimensions, that better fits the conceptual space as captured by the matrix.[7] The more narrow space in the dimensionally reduced matrix will bring distributionally similar tokens closer to one-another in the vector space. The reason for this being that vectors of distributionally similar tokens are no longer orthogonal. LSA is thus only useful if the dimensionality of the reduced matrix, k, is lower than the original matrix's dimensionality n, i.e. $k \ll n$. However, if k is too large it will not capture the underlying semantic structure, and, if k is too small too much information will be lost.

It may be pertinent to add that, as initially being a document-by-term matrix, latent semantic analysis does not capture sequential influences, i.e. syntagmatic relations, among words as they occur in the natural flow of a text. In practice LSA goes from a bag-of-words to a bag-of concepts representation,[8] and in the wake of LSA several other models of latent semantics have been proposed. Some of the more notable are Hyperspace Analogue to Language (Lund *et al.* 1995, Burgess 1998) and Random Indexing (Kanerva *et al.* 2000, Sahlgren 2001)

Random Indexing

Random Indexing is given a more thorough walk-through in Paper 5 in this thesis, and an excellent introduction is given in (Sahlgren 2005). I will, however, attempt a short recapture here. Basically, the construction of context vectors using Random Indexing (RI) can be viewed as a two-step operation.

First, each token in the data is assigned a unique and (usually) randomly generated sparse, high-dimensional, and ternary index vector (random label).[9] Their dimensionality (d) is usually chosen to be in the range of a couple of hundred up to several thousands, depending on the size and redundancy of the data. They consist of a very small number, usually about 1-2%, of randomly distributed +1s and -1s, with the rest of the elements of the vectors set to 0.

Next, the actual context vectors are produced by scanning through the text and each time a token w_i occurs within a sliding context window focused on token w_j, the d-dimensional random label of w_i is added to the context vector for the token w_j, see Figure 3.1. This means that all tokens that appear within the context window of w_j contribute to some degree with their random labels to the context vector for w_j. Usually the proximity is used as a weighting function, thereby letting words closer in context have a higher impact on the semantic representation of a word,

[7] This is often done by singular value decomposition (Forsythe *et al.* 1977), but other methods for dimensionality reduction exist.

[8] Though it has been argued that the meaning of, for instance, a passage of text may be represented independently of the order of its constituting words (Landauer *et al.* 1997), and admittedly LSA has been empirically validated in numerous NLP tasks.

[9] In practise these are generated on-the-fly whenever a never before seen token is encountered in the sliding context window during indexing.

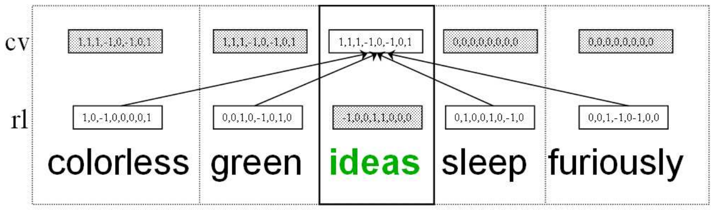


Figure 3.1: A Random Indexing context window focused on the token "ideas", taking note of the co-occurring tokens. The row marked as `cv` represents the continuously updated *context vectors*, and the row marked as `rl` the static randomly generated *index labels* (random labels), which in practice act as addable meta words. Grayed-out fields are not involved in the current token update.

mirroring e.g. dependencies between words of different word classes.[10] Words are thus effectively represented by d-dimensional context vectors that are the (weighted) sum of the random labels of the co-occurring words.

Apart from being a linguistically appealing approach to statistical lexical semantics in that it allows for the encoding of not only paradigmatic but also syntagmatic relations between words, Random Indexing also offers a set of scalability benefits, making it an attractive alternative to e.g. Latent Semantic Analysis (LSA) or Hyperspace Analogue to Language (HAL).

Fistly, RI is inherently incremental. This means, at least in theory, that the context vectors accumulated during indexing can be used for calculating similarity even after only a few word instances have been encountered. In practice the initially very sparse vectors do not carry enough co-occurrence information to be reliable after only a few occurrences of each term (word). On the other hand, with LSA the entire data needs to be processed and represented in a huge matrix, which is then dimensionally reduced before any similarity measures can be performed.[11]

Secondly, RI uses a prefixed dimensionality. This means that when encountering new data the dimensionality of the context vectors does not increase, and it only adds to the number of vectors represented in the model when running into a term never encountered before. This radically lessens the scalability issues with many other word space methods. In HAL, for example, a matrix is created of size $T \times T$, where T is equal to the number of unique terms found in the data set, so far. In LSA, instead a $D \times T$ matrix is built, where D is the number of indexed documents and T, again, is the number of unique terms encountered.

[10] Examples of this is adjectives and verbs giving certain nouns latent properties like "colored", "animate" or "able to speak".

[11] It is possible to fold new documents into the reduced LSA matrix, however, this is a computationally costly operation which makes it unwieldy for large corpora.

Thirdly, RI uses no explicit dimensional reduction. Since the fixed dimensionality of the context vectors is much lower than the number of unique terms in the data, which in turn only governs the number of context vectors, this leads to a significant gain in memory consumption and processing time compared to LSA. In HAL, if one so desires, dimensional reduction can be accomplished by only keeping those dimensions in the matrix that show a high variance (Lund *et al.* 1995), but again at a computational cost.

3.4 Holistic Summarization

The traditional way to perform extractive summarization is to rank the sentences in a source document for their respective appropriateness for inclusion in a summary of this document. These sentences are then concatenated into a summary, which is delivered to the user. This conjugate is seen as containing the sentences most central to the topic of the input text, thus being a representative summary. As contrast, the idea behind a holistic view on summarization is that summaries should be weighed for fitness as a whole, already in the production step. This means that no prejudice is exercised on individual sentences – all sentences are treated as equal. Instead it is their ability to co-operate in forming an overview of the source text that is judged upon.

In order to evaluate this fitness we need to have some way of comparing a source document with one or more summary candidates for content similarity. This is accomplished by letting the concepts we have accumulated by tracking co-occurrences of words in contexts, i.e. by use of Random Indexing, form document signatures. Analogously to how we projected the words' semantic representations into a concept space we can now, by letting the document/summary descriptions be the aggregate of the concept vectors of the words contained in that document/summary, project the documents into a multi-dimensional document space, here represented by the matrix $M_{d,cf}$ below.

$$M_{d,cf} = \begin{pmatrix} w_{d0,1}, & w_{d0,2}, & w_{d0,3}, & \dots, & w_{d0,n} \\ w_{s1,1}, & w_{s1,2}, & w_{s1,3}, & \dots, & w_{s1,n} \\ w_{s2,1}, & w_{s2,2}, & w_{s2,3}, & \dots, & w_{s2,n} \\ w_{s3,1}, & w_{s3,2}, & w_{s3,3}, & \dots, & w_{s3,n} \\ \dots, & \dots, & \dots, & \dots, & \dots \\ w_{sm,1}, & w_{sm,2}, & w_{sm,3}, & \dots, & w_{sm,n} \end{pmatrix}$$

Following the notation above $w_{x,y}$ denotes the weight of "feature" y in the content vector of document/summary x, where m is the number of possible summaries and n the predefined dimensionality as set in the Random Indexing phase. As with the concepts we can now measure the semantic similarity between the document being summarized and a proposed summary by taking the cosine angle between the content vectors of the two.

Here it might well be noted that this optimization of semantic similarity between the source document and the considered summary is not in any way constrained to computationally generated summaries. The summaries being evaluated and selected from could in practice be produced by any means, even being man-made.

3.5 The HolSum Summarizer

HolSum (Hassel and Sjöbergh 2005, 2006, 2007) is a text summarizer that aims at providing overview summaries by generating a set of summary candidates, from which it presents the summary most similar to the source to the user. In practice this means that HolSum tries to represent the various (sub)topics of the source text in the summary to the same extent as they occur in the source text.

HolSum is trainable for most languages provided a splitter that can split the text into the desired extraction segments (e.g. sentences) and tokens suitable for co-occurrence statistics (e.g. words), as defined by that language. Apart from the obligatory sentence and word splitter, and the optional stopword filter and stemmer, there are three main areas of interest in the HolSum system. These three areas, marked with the numbers one through three in the HolSum system layout (see Figure 3.2), are the acquisition semantic knowledge, the application of the acquired semantic knowledge and, lastly, the semantic navigation of the space of possible summaries.

(1) The acquisition of semantic knowledge is carried out by using Random Indexing to build concept representations – context vectors – for a very large vocabulary. Even though we have chosen to use RI as means for acquiring the co-occurrence statistics on words, you could basically use any sufficient model for acquiring such semantic knowledge. It should not, at least in theory, make any difference if you use for instance RI, LSA or HAL. These models are equally language independent given a definition of what constitutes a meaning bearing token (i.e. a "content word") in a specific language, and do all use vectors as their representation for concepts.

(2) In our model document signatures are formed by aggregating context vectors, i.e. the constituting words' co-occurrence vectors. This is not an approach specific to our model, rather it is common practice when using word space models as means for document representation. Nevertheless, it is not, from a linguistic point of view, a particularly appetizing approach to the encapsulation of a documents semantic content, albeit one that clearly improves on the traditional vector space model (Hassel and Sjöbergh 2006). The Achilles' heal of this approach, theory-wise, is that while the formation of concept representations, in Random Indexing, shies away from the bag-of-words approach in that it has the ability to capture syntactically governed semantic relations between words, the document representations regress into a bag-of-concepts model. Even so, relenting to the model we have

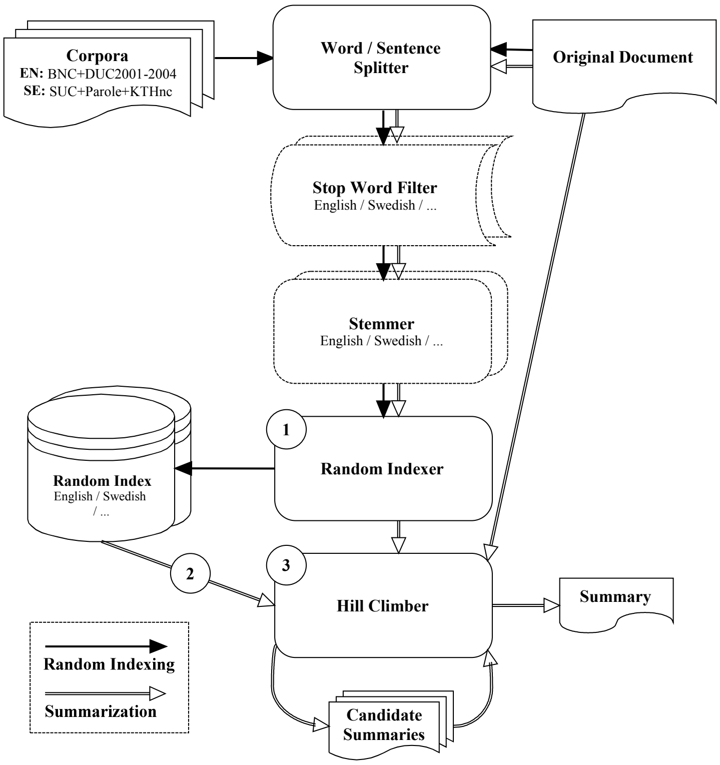


Figure 3.2: A detailed schematic overview of the HolSum system, with its core language independent properties numbered. (1) being the acquisition of semantic knowledge, (2) the application of the acquired semantic knowledge, and (3) the semantic navigation of the space of possible summaries.

there is still room for different models of salience weights when producing these document signatures. In our experiments we have evaluated two such models; the traditional *tf·idf* model and the standard deviation of word occurrences (Hassel and Sjöbergh 2007).

(3) As mentioned, the document signatures crafted in (2), being vectors as they are, can be positioned in a high-dimensional space where they can be compared for content similarity. Lacking a set of man-made summaries to compare and choose from, a suitable set of summaries must be computationally generated. In our current architecture this is performed by using a greedy search algorithm starting in some initial summary, e.g. the leading sentences of the text being summarized, and iteratively reshaping the summary by swapping sentences from the source text in and out of the summary, until no better summary is found (according to some

Azerbaijani President Heydar Aliyev, who is considered the most likely to win the presidential elections, cast his vote today, Sunday, at one of the polling centers near his residence in the center of the capital and took the opportunity to attack his main opponent, Etibar Mammadov. The president, who was elected in September 1993, said in a statement to reporters that "one of the candidates, and you know who I mean, asserts that he has a team and a program, but when the country was on the verge of civil war in 1993, Etibar Mammadov was involved in the political scene so why did he not do anything and why did he not try to stop" the tragedy.

Figure 3.3: Lead summary used as starting point for greedy search (ROUGE-1 37.8%, cosine 0.0310).

Supporters of Azerbaijani President Heydar Aliyev proclaimed today, Monday, that he was reelected for a new term from the first round that took place yesterday, Sunday, while his main opponent Etibar Mammadov, declared that a second round ought to be held. The 4200 polling offices, under the supervision of 180 observers from the Security and Cooperation Organization in Europe, will remain open till 20:00 local time. In order to win in the first round as Aliyev hopes, a candidate must win more than 75% of the votes with a turnout of over 25%.

Figure 3.4: HolSum local maximum summary scoring ROUGE-1 44.0%, with a cosine similarity of 0.995.

criteria). Using this approach it is obvious that we risk getting stranded in a local optimum, however, it is not feasible to exhaustively explore the entire document space in the search of the globally best summary. Furthermore, we do not even know how many "best" summaries there are for the current text, which would be interesting in itself perhaps being a measure of the texts "summaribility", which leaves us with little information on whether we should restart the search or not.

Examples of an "input summary" (lead, 118 words) and the corresponding "end summary" (HolSum output, 94 words) can be found in Figure 3.3 and 3.4. In this particular case the source text these two summaries are derived from has a length of 1588 words, thus yielding the two example summaries a compression rate[12] of 7.4% and 5.9%, respectively. The final summary, as presented to the user, was reached after three iterations of the greedy search.

Stemming in HolSum

Even though stemming as such is an optional preprocessing step prior to building the semantic representations used by HolSum, we have shown that stemming im-

[12] The term "compression rate" is further explained in Section 4.1.

*	Do not remove or replace anything
-	Remove matched if a preceding vowel is found
+	Remove matched
=	Remove matched if matching the whole word
.	Stop matching (break)
abc	Replace with abc

Table 3.1: The set of commands applicable to words being stemmed.

hals	***.**	Do not remove or replace and stop matching. ("neck")
abel	**-**	Remove matched if a preceding vowel is found.
sköt	**+.skjut**	Remove *sköt*, insert *skjut* ("shoot" or "push") and break

Table 3.2: Example set of exception rules.

proves the resulting summaries (Hassel and Sjöbergh 2006). As has been shown in information retrieval, stemming becomes more crucial given a language with richer morphology (Carlberger *et al.* 2001). Two stemmers have been used in the experiments with "holistic" summarization. For English we have used the widely spread Porter stemmer and, for Swedish, the Euroling SiteSeeker stemmer, both to be briefly presented below.

The Porter Stemmer

The Porter stemmer (Porter 1980) is used for English stemming in HolSum. It removes about 60 different suffixes and uses rewriting rules in two steps. The Porter stemmer is quite aggressive when creating stems and does some overstemming. Despite it creating many equivalence classes, it still performs well in many precision/recall evaluations. Also, the Porter stemmer does not handle irregularities at all, which means that irregular forms such as *goose/geese*, *swim/swam* and *conceive/conception* will all be given distinct stems. The Porter stemmer is still one of the most widely used stemmers for English. The version used in the experiments detailed in this thesis is the official Java port (release 4).

The Euroling SiteSeeker Stemmer

The Euroling SiteSeeker stemmer (Carlberger *et al.* 2001) uses about 150 stemming rules for Swedish. It uses a technique where it, with a small set of suffix rules, in a number of steps modifies the original word into an appropriate stem. The stemming is done in (up to) four steps and in each step no more than one rule from a set of rules is applied. This means that 0–4 rules are applied to each word passing through the stemmer. Each rule, in turn, consists of a lexical pattern to match with the suffix of the word being stemmed and a set of modifiers, or commands. For an example of such modifiers, see Table 3.1.

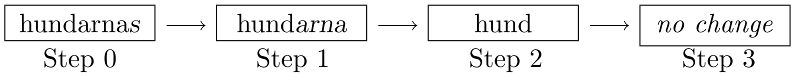


Figure 3.5: Stemming of the word *hundarnas* (“the dogs” ’ *genitive form plural*) to *hund* (“dog”).

As can be seen in the example in Figure 3.5, in step 0 genitive-s and active-s are handled. These are basically -s stripping rules. Definite forms of nouns and adjectives are handled in step 1, as well are preterite tense and past participle. In step 2 mainly plural forms of nouns and adjectives are handled. Noun forms of verbs are handled in step 3. In step 3 there are also some fixes to cover exceptions to previously executed rules, see Table 3.2.

The stem of a word does not have to be of the same part-of-speech as the word; in whatever sense you can talk about part-of-speech regarding stems. The rules are specifically designed so that word classes can be “merged” where appropriate. This means that, for example, *cykel* (“bicycle”) and *cyklade* (“rode a bicycle”) are both stemmed to the common stem *cykl*, thus grouping conceptually related words.

3.6 Chapter Summary

In this chapter we have defined the term *language independent summarization* as a summarization method or application that seamlessly moves from one language or domain to another. This transfer should be facilitated by the systems ability to adapt to the irregularities natural languages exhibit, e.g. by learning new instances of language use, without the use of hard-coded language specific knowledge or need of annotated or structured data.

As we have seen in this chapter there are several linguistically motivated text processing tasks that need to be addressed from a language independence point of view. Among these are such tasks as *tokenization* and *sentence splitting*. Furthermore, we have discussed the impact of these tasks on the representation of the contents of documents, i.e. *document signatures*. The impact of *stemming* and *compound splitting* on document signatures is also discussed. These document signatures can then be compared for similarity using the *vector space model*. Related to this discussion is the notion of *salience* and how one can promote topically relevant words and *concept representations* in the document signatures. These concept representations are crafted by gathering *word co-occurrence statistics* used for grouping semantically related words in a *word space model*. This model is based on the *distributional hypothesis* according to which words occurring in similar contexts tend to have similar *meaning* or *function*.

Lastly, we have presented the notion of *holistic summarization* were a set of summaries are internally ranked and the “best” summary presented to the user, rather than the traditional conjugate of individually ranked sentences. This notion has lead to the instantiation of the *HolSum summarizer*, which employs the *Random Indexing* word space model for crafting concept representations. These concept representations are used to form document signatures for both the *input text* as well as *generated summaries*, which are compared for *semantic similarity* in a *document space*. The discussion up to this point supports the *near language independence* of the approach. However, we have also pointed out the regression from a *syntagmatically distributional model* of concepts to a *bag-of-concepts* representation of documents. We have also seen a brief overview of the *Swedish stemmer* used in HolSum.

Chapter 4

Summarization Evaluation

A crucial phase of the development of any system, method or methodology is the evaluation and validation of said task. Natural Language Processing (NLP) systems are no exception. Rather, given the irregularities of (human) languages it is simply an all too daunting task to logically prove this loosely defined body of possible utterances. A common approach to bridge this fact is to use validation by induction, thus testing against a body of data assumed to be a representative subset of the near infinite complete set of utterances. The prospect of a specific approach can thus be empirically validated.

Most automatic text summarization systems today are extraction-based systems. However, work has been directed towards post-editing of extracted segments, e.g. sentence/phrase reduction and combination, thus at least creating the illusion of abstracting in some sense. This has lead to the situation where evaluation has to tackle comparison of summaries that do not only differ in wording but maybe also in specificity and bias.

Furthermore, in automatic text summarization, as well as in for example machine translation, there may be several equally good summaries (or in the case of MT - translations) for one specific source text, effectively making evaluation against one rigid reference text unsatisfactory. Also, evaluation methods that allow for evaluation at different compression rates should be favored as experiments have shown that different compression rates are optimal for different text types and genres, or even different texts within a text type or genre. The automatic evaluation methods presented in this paper mainly deal with content similarity between summaries and the original document.

Today, there is no single evaluation scheme that provides for all these aspects of the evaluation, so a mixture of methods described in this paper should perhaps be used in order to cover as many aspects as possible, thus making the results comparable with those of other systems, shorten the system development cycle and support just-in-time comparison among different summarization methods. Clearly some sort of standardized evaluation framework is heavily in need in order to ensure replication of results and trustworthy comparison among summarization systems.

4.1 Two Basic Properties

Evaluating summaries and automatic text summarization systems is not a straightforward process. What exactly makes a summary beneficial is an elusive property. Generally speaking, there are at least two properties of the summary that must be measured when evaluating summaries and summarization systems: the Compression Ratio (how much shorter the summary is than the original);

$$CR = \frac{length\ of\ Summary}{length\ of\ Full\ Text}$$

and the Retention Ratio (how much information is retained);

$$RR = \frac{information\ in\ Summary}{information\ in\ Full\ Text}$$

Retention Ratio is also sometimes referred to as Omission Ratio (Hovy 1999). An evaluation of a summarization system should at least in some way address both of these properties. In many scenarios CR is cast aside in preference for a statically defined target summary length that is independent of the source document(s) original length. The retention ratio has thus gained most of the attention.

A first broad division in methods for evaluating automatic text summarization systems, as well as many other systems, is the division into intrinsic and extrinsic evaluation methods (Spärck-Jones and Galliers 1995, Mani and Maybury 1999).

4.2 Intrinsic Evaluation

Intrinsic evaluation measures the summarization system without regard to its target audience. Instead the focus here lies on the production phase of a summary's lifespan. Most summarization evaluation schemes are intrinsic, and are often carried out by comparison to some gold standard. In NLP an "ideal" set of reference examples[1] is often referred to as a gold-standard corpus. A gold standard is usually seen as a model of excellence and acts as the upper bound of what is reasonable to achieve by automated means. In the case of summarization this gold-standard set of summaries can be generated by a reference summarization system or, more often than not, be provided by human informants. Intrinsic evaluation has mainly focused on the coherence and informativeness of summaries, thereby measuring output quality only.

[1] These examples might, depending on the NLP tasks, be grammatical error types, word sense classes or part-of-speech tags etc.

Summary Coherence

Summaries generated through extraction-based methods (cut-and-paste operations on phrase, sentence or paragraph level) sometimes suffer from parts of the summary being extracted out of context, resulting in coherence problems (e.g. dangling anaphora or gaps in the rhetorical structure of the summary). One way to measure this is to let subjects rank or grade summary sentences for coherence and then compare the grades for the summary sentences with the scores for reference summaries, with the scores for the source sentences, or for that matter with the scores for other summarization systems.

Summary Informativeness

One way to measure the informativeness of the generated summary is to compare the generated summary with the text being summarized in an effort to assess how much information from the source that is preserved in the condensation. Another is to compare the generated summary with a reference summary, measuring how much information in the reference summary that is present in the generated summary. For single documents traditional precision and recall figures can be used to assess performance as well as utility figures and content-based methods (see below).

Sentence Precision and Recall

Sentence recall measures how many of the sentences in the reference summary that are present in the generated summary. Analogously, precision can in this case be defined as the number of sentences in the generated summary that are present in the reference summary. Precision and recall are standard measures for Information Retrieval and are often combined in a so-called F-measure (Van Rijsbergen 1979). The main problems with these measures for text summarization is that they are not capable of distinguishing between many possible but equally efficacious summaries, and that summaries that differ quite a lot content-wise may get very similar scores (Mani 2001).

Sentence Rank

Sentence rank is a more fine-grained approach than precision and recall (P&R), where the reference summary is constructed by ranking the sentences in the source text by worthiness of inclusion in a summary of the text. Correlation measures can then be applied to compare the generated summary with the reference summary. As in the case of precision and recall this method mainly applies to extraction-based summaries, even if standard methods of sentence alignment with abstracts can be applied (Marcu 1999, Jing and McKeown 1999). It is, however, not a particularly natural task for a human and one might suspect mimicking a computer algorithm is not the best way to collect reference summaries.

The Utility Method

The Utility Method (UM) (Radev *et al.* 2000) allows reference summaries to consist of extraction units (sentences, paragraphs etc.) with a "fuzzy" membership in the reference summary. In UM the reference summary contains all the sentences of the source document(s) with confidence values, ranging from 0 to 10, for their inclusion in a summary. As in the case of Sentence Rank, these confidence values are assigned by human informants. Furthermore, the UM methods can be expanded to allow extraction units to exert negative support on one another. This is especially useful when evaluating multi-document summaries, where in case of one sentence making another redundant it can automatically penalize the evaluation score, i.e. a system that extracts two or more "equivalent" sentences gets penalized more than a system that extracts only one of the aforementioned sentences and a, say, less informative sentence (i.e. a sentence that has a lower confidence score).

In contrast to precision/recall and Percent Agreement, which is defined as the number of observed agreements over the total number of possible agreements (Gale *et al.* 1992), UM allows summaries to be evaluated at different compression rates. UM is, like many similar evaluation metrics, mainly useful for evaluating extraction-based summaries. More recent evaluation experiments has led to the development of the Relative Utility metric (Radev and Tam 2003).

Content Similarity

Content similarity measures (Donaway *et al.* 2000) can be applied to evaluate the semantic content in both extraction-based summaries and true abstracts. One such measure is the Vocabulary Test (VT) where standard Information Retrieval methods (see Salton and McGill 1983) are used to compare term frequency vectors calculated over stemmed or lemmatized summaries (extraction-based or true abstracts) and reference summaries of some sort. Controlled thesauri and "synonym sets" created with Latent Semantic Analysis (Landauer *et al.* 1998) or Random Indexing (Sahlgren 2005), see Section 3.3, can be used to reduce the terms in the vectors by combining the frequencies of terms deemed synonymous, thus allowing for greater variation among summaries. This is especially useful when evaluating abstracts.

One disadvantage of these methods is that they are quite sensitive to negation and word order differences that may affect the interpretation of the content. A possible approach to overcome this is to use frequencies on sequences of words (terms), i.e. n-grams, instead of on single terms (see Section 4.4).

Also, with Latent Semantic Analysis or Random Indexing one must be aware of the fact that these methods do not necessarily produce true synonym sets, these sets typically also include antonyms, hyponyms and other terms that occur in similar semantic contexts (on word or document level for RI and document level for LSA). These methods are however useful for extraction-based summaries where little rewriting of the source fragments is done, and when comparing fragmentary summaries, such as key phrase summaries.

4.3 Extrinsic Evaluation

Unlike intrinsic evaluation, extrinsic evaluation put the focus on the end user. It thus measures the efficiency and acceptability of the generated summaries in some task, for example relevance assessment or reading comprehension. Also, if the summary contains some sort of instructions, it is possible to measure to what extent it is possible to follow the instructions and the result thereof. Other possible measurable tasks are information gathering in a large document collection, the effort and time required to post-edit the machine generated summary for some specific purpose, or the summarization system's impact on a system of which it is part of, for example relevance feedback (query expansion) in a search engine or a question-answering system.

Several game like scenarios have been proposed as surface methods for summarization evaluation inspired by different disciplines, among these are The Shannon Game (information theory), The Question Game (task performance), The Classification/Categorization Game and Keyword Association (information retrieval).

The Shannon Game

The Shannon Game, which is a variant of Shannon's measures in Information Theory (Shannon 1951), is an attempt to quantify information content by guessing the next token, e.g. letter or word, thus recreating the original text. The idea has been adapted from Shannon's measures in Information Theory where you ask three groups of informants to, letter by letter or word by word, reconstruct important passages from the source article having seen either the full text, a generated summary, or no text at all. The information retention is then measured in number of keystrokes it takes to recreate the original passage. Hovy (see Hovy and Marcu 1998) has shown that there is a magnitude of difference across the three levels (about factor 10 between each group). The problem is that Shannon's work is relative to the person doing the guessing and therefore implicitly conditioned by the reader's knowledge. The information measure will infallibly change with more knowledge of the language, the domain, etc.

The Question Game

The purpose of the Question Game is to test the readers' understanding of the summary and its ability to convey key facts of the source article. This evaluation task is carried out in two steps. First the testers read the source articles, marking central passages as they identify them. The testers then create questions that correspond to certain factual statements in the central passages. Next, assessors answer the questions 3 times: without seeing any text (baseline 1), after seeing a system generated summary, and after seeing original text (baseline 2). A summary successfully conveying the key facts of the source article should be able to answer most questions, i.e. being closer to baseline 2 than baseline 1. This evaluation scheme has for example been used in the TIPSTER SUMMAC text summarization evaluation Q&A[2] task, where Mani *et al.* (1998) found an informativeness ratio of accuracy to compression of about 1.5.

The Classification Game

In the classification game one tries to compare classifiability by asking assessors to classify either the source documents (testers) or the summaries (informants) into one of N categories. Correspondence of classification of summaries to the originals is then measured. An applicable summary should be classified into the same category as its source document. Two versions of this test were run in SUMMAC (Mani *et al.* 1998). If one would define each category by one or more keywords the classification game generalizes into a keyword association scenario.

Keyword Association

Keyword association is an inexpensive, but somewhat shallower, approach that relies on keywords associated (either manually or automatically) to the documents being summarized. For example Saggion and Lapalme (2000) presented human judges with summaries generated by their summarization system together with five lists of keywords taken from the source article as presented in the publication journal. The judges were then given the task to associate the each summary with the correct list of keywords. If successful the summary was said to cover the central aspects of the article since the keywords associated to the article by the publisher were content indicative. Its main advantage is that it requires no cumbersome manual annotation.

4.4 Evaluation Tools

In order to allow a more rigorous and repeatable evaluation procedure, partly by automating the comparison of summaries, it is advantageous to build an extract corpus containing originals and their extracts, e.g. summaries strictly made by extraction of whole sentences from an original text. Each extract, whether made by a human informant or a machine, is meant to be a true summary of the original, i.e. to retain the meaning of the text to as high degree as possible. Since the sentence units of the original text and the various summaries are known entities, the construction and analysis of an extract corpus can almost completely be left

[2] Question and Answering; a scenario where a subject is set to answer questions about a text given certain conditions, for example a summary of the original text.

to computer programs, if these are well-designed. A number of tools have been developed for these purposes.

Summary Evaluation Environment

Summary Evaluation Environment (SEE; Lin 2001) is an evaluation environment in which assessors can evaluate the quality of a summary, called the peer text, in comparison to a reference summary, called the model text. The texts involved in the evaluation are preprocessed by being broken up into a list of segments (phrases, sentences, clauses, etc.) depending on the granularity of the evaluation. For example, when evaluating an extraction-based summarization system that works on the sentence level, the texts are preprocessed by being broken up into sentences.

During the evaluation phase, the two summaries are shown in two separate panels in SEE and interfaces are provided for assessors to judge both the content and the quality of summaries. To measure content, the assessor proceeds through the summary being evaluated, unit by unit, and clicks on one or more associated units in the model summary. For each click, the assessor can specify whether the marked units express all, most, some or hardly any of the content of the clicked model unit. To measure quality, assessors rate grammaticality, cohesion, and coherence at five different levels: all, most, some, hardly any, or none. Quality is assessed both for each unit of the peer summary and for overall quality of the peer summary (coherence, length, content coverage, grammaticality, and organization of the peer text as a whole). Results can, of course, be saved and reloaded and altered at any time.

A special version of SEE 2.0 has for example been used in the Document Understanding Conferences (DUC)[3] evaluation campaigns 2001–2004 (Harman and Marcu 2001) for intrinsic evaluation of generic news text summarization systems (Lin and Hovy 2002). In DUC 2001 the sentence was used as the smallest unit of evaluation.

MEADeval

MEADeval (Winkel and Radev 2002) is a Perl toolkit for evaluating MEAD- and DUC-style extracts, by comparison to a reference summary (or "ideal" summary). MEADeval operates mainly on extract files, which describe the sentences contained in an extractive summary: which document each sentence came from and the number of each sentence within the source document – but it can also perform some general content comparison. It supports a number of standard metrics, as well as some specialized (see table 4.1). In this table the normalized precision and recall are normalized by the length, in words, of each sentence, and the simple

[3] The Document Understanding Conferences is an ongoing series of evaluation campaigns, funded by the Advanced Research and Development Activity and run by the National Institute of Standards and Technology, aimed at pushing the scientific boundaries of summarization and to enable researchers to participate in comparable large-scale experiments.

Extracts only	General text
precision	unigram overlap
recall	bigram overlap
normalized precision	cosine
normalized recall	simple cosine
kappa (inter-rater agreement)	
relative utility	
normalized relative utility	

Table 4.1: Metrics supported by MEADeval.

cosine is without adjustments for inverse document frequency (idf). The Relative Utility and Normalized Relative Utility metrics are described in Radev and Tam (2003), see also Section 4.2.

A strong point of Perl, apart from platform independency, is the relative ease of adapting scripts and modules to fit a new summarization system. MEADeval has, for example, been successfully applied to summaries generated by a Spanish lexical chain summarizer and the SweSum[4] summarizer in a system-to-system comparison against model summaries (Alonso Alemany and Fuentes Fort 2003).

ISI ROUGE – Automatic Summary Evaluation Package

The ISI ROUGEeval package Lin (2003), later just referred to as ROUGE, is an attempt at automating the evaluation of summaries, which measures word n-gram co-occurrences between summary tuples. These tuples usually consist of one or more system generated summaries as well as one or most often several hand-crafted reference summaries, which act as a gold standard to which the generated summaries are compared. ROUGE, short for Recall-Oriented Understudy for Gisting Evaluation, is an adaption of the IBM BLEU score for Machine Translation (Papineni *et al.* 2001, NIST 2002).

ROUGE started as recall oriented, in contrast to the precision oriented BLEU script, and separately evaluates various word n-gram measurements. Also, ROUGE does not apply any length penalty (brevity penalty), which is natural since text summarization involves compression of text and thus rather should reward shorter extract segments as long as they score well for content. ROUGE has since its early versions been equipped with precision as well as F-measures, which combines precision and recall into one (optionally biased) metric. ROUGE as of version 1.5.5 scores for the following:

- ROUGE-1…n: Word n-gram overlap between the system summary and the reference summaries

[4] SweSum mainly being a Swedish language text summarizer, also supports plug-in lexicons and heuristics for other languages, among these Spanish.

- ROUGE-L: Longest common word subsequence between the system summary and the reference summaries
- ROUGE-W: As ROUGE-L, but weighted to promote consecutive words
- ROUGE-Sn: Skip-bigram co-occurrence statistics without gap length limit and with maximum gap lengths of n words
- ROUGE-SUn: As ROUGE-Sn, but without unigram counting

With ROUGE one can also, optionally, use Porter stemming and stopword filtering prior to computing the measures. This option is unfortunately only profitable for evaluation on English if one does not venture replacing these with equivalents for other languages.

In addition to the n-gram overlap metrics ROUGE can also, optionally, account for Basic Elements (Hovy *et al.* 2005). These minimal semantic units are defined as the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases), expressed as a single item; or a relation between a head-BE and a single dependent, expressed as a triple {head|modifier|relation}. This measure is, as the use of stemming and stopword filtering, language specific and requires a syntactic parser as well as a set of "cutting rules" to extract just the valid BEs from the tree.

ROUGE has been verified for extraction-based summaries with a focus on content overlap. According to in-depth studies based on various statistical metrics and comparison to the results of several DUC runs this evaluation metric does seem to correlate well with human evaluation (Lin and Hovy 2002, 2003b,a, Hovy *et al.* 2005). However, it is quite easy to maliciously fabricate a summary that optimizes ROUGE scores nearly beyond human agreement, but that at a mere glance would be rejected by the human eye (Sjöbergh 2007). Completely automated summarization evaluation using ROUGE is thus not yet within reach.

Despite some criticism ROUGE scores have in recent years become a *de facto* standard in the evaluation of summarization systems, at least for English, and have been used in the DUC campaigns running from 2004 through (at least) 2007.

4.5 The KTH News and Extract Corpora

The HolSum summarizer, see Section 3.5, has been evaluated both using English and Swedish training and evaluation data. The motivation for evaluating also on English, despite the fact that there already exists a score of summarizers geared at this language, is that exactly this fact gave us the opportunity to compare our results with those of other systems on well-acknowledged corpora using established evaluation metrics.[5] However, as an integral part of any research, the evaluation of resource lean and portable summarization methods should preferably share these

[5] The experiments on English were carried out using the British National Corpus (Burnard 1995) and the document sets from the Document Understanding Conferences (DUC) evaluation campaigns 2001–2004 (DUC 2007). The English evaluation was performed on the DUC data using ROUGEeval-1.4.2 (see Section 4.4), mimicking the evaluation set-up for task 2 in DUC 2004 (Over and Yen 2004).

properties in order to also bring evaluation to the new domain. A series of such attempts have led to the development of several methods for the collection of corpora, ranging from a large-scale corpus aimed at machine learning and data mining to more modest corpora aimed explicitly at evaluation.

The KTH News Corpus

As has already been established in previous chapters, the HolSum summarizer demands substantial amounts of (unannotated) natural language training data. In order to facilitate the demand of large scale corpora for the Swedish domain, for training as well as evaluation, we developed a web spider – newsAgent – specifically targeted at on-line editions of major Swedish news papers, business and computer magazines, as well as press releases from humanitarian organizations and the Swedish parliament (Hassel 2001a).

This web spider uses hand crafted regular expression filters that remove everything but the actual body of the news text. Once these filters are in place the system is practically care-free, at least until the source site is redesigned. This does in practice occur quite seldom since most site owners have identified their visitors' need to feel "at home". The newsAgent spider has, as a part of the Business Intelligence tool NyhetsGuiden[6] (see Hassel 2001b), under the duration of approximately one and a half year collected news stories comprising some 13 million words (200,000 articles). By polling the different sources every fifth minute, recording such meta-data as time of publication, a temporal impression of the news flow is achieved. This corpus is further discussed in Paper 1.

The KTH News Corpus naturally forms a material part of the Swedish language model used by HolSum, but has also spawned several subcorpora used for various evaluation tasks. One such corpus is the Swedish part of the KTH eXtract Corpus, see below.

The KTH eXtract Corpus

To facilitate intrinsic summarization evaluation in a new domain, e.g. on a new language, we have developed a suite of web-based tools for the collection of extractive summaries provided by human informants. The KTH eXtract Corpus (KTHxc), discussed in Paper 4, contains a number of original texts for which the informant is assisted in the construction of extraction-based summaries. In this task the KTHxc Collector guides the informant in creating a summary in such a way that only full extract units (most often sentences) are selected for inclusion in the summary. The interface allows for the reviewing of sentence selection at any time, as well as reviewing of the constructed summary before submitting it to the corpus.

Once the extract corpus is compiled, the KTHxc Browser allows for navigating and viewing the corpus, as well as exporting extracts in various formats, e.g.

[6] "NewsGuide" in English.

the format SEE uses for human assessment (see Section 4.4). Semi-automatic evaluation can also be conducted directly in the interface in the sense that the inclusion of sentences in the various extracts for a given source text can easily be compared to a submitted system generated summary. This allows for a quick adjustment and evaluation cycle in the development of an automatic summarizer. One can, for instance, adjust parameters of the summarizer and directly obtain feedback of the changes in performance, instead of having a slow, manual and time consuming evaluation.

The KTHxc Collector gathers statistics on how many times a specific extract unit from a text has been included in a number of different summaries. Thus, an "ideal" summary, or reference summary, can be composed using only the most frequently chosen sentences. Further statistical analysis can evaluate how close a particular extract is to the reference summary. This approach, however, suffers from not being able to model inter-sentential coherence dependencies (Hassel and Dalianis 2005). With the current lack of co-selection statistics the approach currently used is a simple sentence precision-recall comparison with all submitted extracts (Hassel and Sjöbergh 2005).

Since the relatively few user instructions are separated from the actual source code the KTHxc tool-kit can with relative ease be ported to other languages. So far corpus collection and evaluation has been conducted for Swedish as well as Danish and English (Hassel 2003, Hassel and Dalianis 2005). The University of Bergen has initiated a similar effort for Norwegian and has developed some similar tools (Dalianis *et al.* 2004).

4.6 Chapter Summary

In this chapter we have established the need for *empiric evaluation* in NLP, given the *highly irregular* nature of natural languages. This is usually achieved by means of *induction*, by evaluating the system on *representative data*. The possibility that several summaries of the same text might be *equally efficacious* was also identified as a concern in summarization evaluation, a concern often tackled with the use of several *reference summaries*.

Also identified was two basic properties of summaries, where the *compression ratio* is defined as how much shorter the summary is than the original text, and the *retention ratio* defined as how much information from the original text that is retained in the summary. A division into two major approaches to evaluation was presented. The first approach, *intrinsic evaluation*, measures the summary with little regard to its intended use, while the other, *extrinsic evaluation*, measures the summary's efficiency in some task. A survey of *evaluation schemes* was given for each of the two.

After a survey of some of the available summarization tools, amongst these the reputable *ISI ROUGE*, an overview of the two corpora collection and evaluation environments developed as part of the summarization framework laid forth in this thesis was given. These two systems are the *KTH News Corpus*, for the collection of large-scale corpora used for language modeling, and the *KTH eXtract Corpus*, for the collection of gold-standard extracts used for summary evaluation.

Chapter 5

Conclusion

There is an end to everything, to good things as well (Geoffrey Chaucer, 1343–1400)

5.1 Overview and Conclusions of Included Papers

In this section I will briefly comment on the papers included in this thesis. The main purpose is to give them a context by elaborating on motivation, execution and effect. For those papers that have been written in conjunction with others, I will also declare my part in said paper.

Paper 1.

Internet as Corpus – Automatic Construction of a Swedish News Corpus (Hassel 2001a)

In order to evaluate automatic summarizers or information extraction and retrieval tools, but also to train these tools to make their performance better, one needs to have a corpus. For English and other widespread languages there are freely available corpora, but this is not necessarily the case for Swedish (depending on the NLP task). Therefore we needed to collect a balanced corpus mainly consisting of news text in Swedish. We used the Internet as our source. In total we automatically collected approximately 200,000 news articles between May 2000 to June 2002 containing over 13 million words. The news texts collected were news flashes and press releases from large Swedish newspapers like Svenska Dagbladet, Dagens Nyheter and Aftonbladet, business and computer magazines, as well as press releases from humanitarian organizations like Amnesty International and RFSL,[1] and authorities like Riksdagen.[2]

[1] Riksförbundet För Sexuellt Likaberättigande; A gay, lesbian, bisexual and transgendered lobby organization.

[2] The Swedish Parliament.

The KTH News Corpus (KTHnc) has since been used to train a Named Entity tagger (Dalianis and Åström 2001), that has been evaluated as part of the SweSum text summarizer (Paper 3) and to train the HolSum summarization system for the evaluation on Swedish (Paper 5).

Furthermore, the corpus has also been used for evaluation of the impact of Swedish stemming in a traditional search engine (Carlberger *et al.* 2001), evaluating grammatical error detection rules (Knutsson 2001), bootstrapping a free part-of-speech lexicon (Sjöbergh 2003), evaluation of the use of stemming, compound splitting and noun phrase chunking in document clustering (Rosell 2003, Rosell and Velupillai 2005) and for evaluating an unsupervised method to find errors in text using chunking (Sjöbergh 2005), to name but a few.

Paper 2.

Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools (Dalianis and Hassel 2001)

SweSum, the first automatic text summarizer for Swedish news text, was constructed in 1999 (Dalianis 2000). SweSum works in the text extractor paradigm. This means that it extracts the most significant parts, in this case sentences, from a text and by concatenating them creates a new shorter non-redundant text. This paradigm is currently the most common among automatic text summarizers.

We first made an intrinsic (see section 4.2) qualitative subjective evaluation of SweSum using the techniques described by Firmin and Chrzanowski (1999). Our informants were students at our Human Language Technology course. The students were instructed to judge the summarized texts, by ocular inspection, and decide if the text in question was perceived as well formed in terms of coherence and content. That is, the students rated the SweSum generated summaries for summary coherence (see Section 4.2) and summary informativeness (see Section 4.2). We found that the coherence of the text was intact at 30 percent compression rate and that the information content was intact at 25 percent compression rate.

The following year we improved our experiment by making a more objective extrinsic (see Section 4.3) evaluation of the text summarizer. This time we used 100 manually annotated news texts and corresponding queries (Carlberger *et al.* 2001). Again we instructed students attending our Human Language Technology course to execute SweSum with increasing compression rates on the 100 annotated texts, in an effort to find answers to the predefined questions in a Question Game-like scenario (see Section 4.3). The results showed that at a 40 percent compression rate the correct answer rate, as given by the informants, was 84 percent. Both these methods needed a large human effort, a more efficient evaluation framework was clearly in demand.

Regarding this technical report; I set up the experiments and Hercules Dalianis and I took equal parts in their execution, the interpretation of the results and the writing of the report.

Paper 3.

Exploitation of Named Entities in Automatic Text Summarization for Swedish (Hassel 2003)

In (Dalianis and Åström 2001) a Named Entity recognizer called SweNam was constructed for Swedish named entity recognition. Named entity recognition is the method that from a text extracts names of persons, organizations, geographical and geopolitical locations, and possibly also expressions of times, quantities, monetary values, percentages etc. SweNam was trained on the KTH News Corpus. We were keen on finding out if named entity recognition could improve automatic text summarization. Therefore we connected the original SweSum summarizer with SweNam, where SweNam acted as a preprocessor to SweSum.

We were not completely satisfied with our extrinsic Question and Answering scheme (see Paper 2) in evaluating our text summarizer and wanted to bring evaluation a step further to a more intrinsic approach. Therefore we created the KTH eXtract Corpus (KTHxc), a corpus of manual extracts from original texts that could be used as a gold standard. A gold-standard summary, or "ideal" extract summary, can then repeatedly be compared with automatic summaries generated by SweSum. A group of human informants were presented news articles one at a time, in random order, so that they could select sentences for extraction. The submitted extracts were allowed to vary between 5 and 60 percent of the length of the source text. The advantage of having extracts was that we could directly compare what humans selected as informative or good sentences to include in an extract summary with what the machine, i.e. SweSum, selected. Different settings in, and incarnations of, an extractive summarizer can thus be easily compared. Even though the continuous growth of the corpus is necessary in order to avoid overfitting, the effort of collecting the corpus and the repeated use of it in evaluation is still far less than in previous attempts.

The results of one such evaluation showed that named entities tend to prioritize sentences with a high information level on the categories used. They tend to prioritize elaborative sentences over introductory and thus sometimes are responsible for serious losses of sentences that give background information. Our finding were that named entity recognition must be used with consideration so it will not make the summary too information intense and consequently difficult to read. Also, it may actually in extreme cases lead to condensation of redundancy in the original text. One tactic to counter this phenomenon could of course be to restrain the extent to which each named entity is used in the ranking of the individual sentences.

Paper 4.

Generation of Reference Summaries (Hassel and Dalianis 2005)

When developing text summarizers and other information extraction tools it is extremely difficult to assess the performance of these tools. One reason for this is that evaluation is time-consuming and often requires considerable manual efforts. When changing the architecture of the summarizer one needs to once again undergo the costly evaluation process. Therefore it is fruitful to have an environment in which one directly can assess the result from a text summarizer repeatedly and automatically.

In this paper we present an integrated web-based system for the collection of extract-based corpora, and for evaluation of summaries and summarization systems. The system assists in the collection of extractive summaries provided by human informants by guiding the user in creating a summary in such a way that only full extract units (most often sentences) are selected for inclusion in the summary. The informant is given visual feedback of the selection, and can at any time review the resulting summary before submitting it. When the extract corpus is in place it can be used repeatedly with little effort. An advantage is that one can easily create an extract corpus in any language, since the user interface is separate from the core language independent functionality,[3] and evaluate any text summarizer, as long as the statistics provided by the informants is reliable.

Also, the system allows for the generation of "reference" summaries by majority vote. However, one drawback of the approach is that in a situation of low agreement among the informants the corpus gives unduly favors to summarization systems that use sentence position as a central weighting feature, since this also is a tie-breaking scheme when creating these reference summaries.

Another caveat is that while the resulting reference summaries might well be efficacious, in its current incarnation the system views the sentences voted upon as a bag-of-sentences, presupposing their independence of one another. A more theoretically solid approach, in contrast to positional tie-breaking, would be to make use of the co-occurence of selections within each submitted extract in order to model inter-segmental dependencies.

This paper is in part based on the collaborative work in the ScandSum network (Dalianis *et al.* 2003, 2004, de Smedt *et al.* 2005). The experiments detailed in (Hassel and Dalianis 2005) have been carried out within the framework of ScandSum, however, all of the implementation and most of the data collection and evaluation detailed therein have been conducted by myself, as well as the writing and presentation of the paper.

[3] This property is often referred to as "skinned" interfaces.

Paper 5.

Navigating Through Summary Space – Selecting Summaries, Not Sentences (Hassel and Sjöbergh, submitted 2007)

Going more application oriented this article, a conjugate of three conference papers (Hassel and Sjöbergh 2005, 2006, 2007), outlines a series of experiments pertaining to near language independent summarization. The aim was to develop a summarization method that, with the use of only a very few basic language tools, can be quickly assembled for basically any language that lack large bodies of structured or annotated resources or advanced tools for linguistic analysis. Along these lines we here present and evaluate a novel method for extraction-based summarization that attempts to give an overview by, from a set of summary candidates, selecting the summary most similar to the source text. It accomplishes this by comparing whole summaries at once, not, as traditionally is done in extractive summarization, by ranking individual extraction segments (see Section 3.4).

For this purpose we employ statistical lexical semantics in order to model the semantic contents of a text and its summary, respectively. The overall impact of the summary is then calculated, making no judgments on individual sentences. A simple greedy search strategy – hill-climbing – is then used to search through a space of possible summaries, generated by evaluating permutations of sentence subsets extracted from the source text. Starting the search with the leading sentences of the source text has proven to be a powerful heuristic, but other search strategies are also evaluated.

The method is evaluated on a corpus of Swedish extracts provided by informants. On this data it performed poorly compared to a traditional extraction-based summarizer, SweSum. The main reason for this being due to the fact that our method tries to cover all topics represented in the original text, to the same proportion, and these man-made extracts were not produced to reflect the whole contents of the texts, but rather to cover only the main topic. It does, however, perform well on short extracts derived from fairly long news texts when compared to man-made summaries, such as those used in the DUC 2004 summarization evaluation campaign. On this task the proposed method performs better than several of the systems evaluated on the same data, but worse than the best systems.

In conclusion, even though the HolSum approach does not outperform the best systems for English it is trivial to port to other languages. It also has the intuitively appealing property of optimizing semantic similarity between the generated summary and the text being summarized. Also, it should be noted that this property is not in any way constrained to extractive summarization, even though we here use it to differentiate between extractive summaries. The summaries being evaluated and selected from could in practice be generated by any means, even being man-made.

This journal paper (submitted) is based on a set of three previously published conference papers (see below). Apart from having the main responsibility for compiling and rewriting these three papers into one longer coherent article, my part in each of the papers has been:

- **Towards Holistic Summarization: Selecting Summaries, not Sentences** (Hassel and Sjöbergh 2006). On this Jonas Sjöbergh did the main part of the implementation, myself and Sjöbergh took equal part in the experiments, and I did the main part in writing the paper.
- **A Reflection of the Whole Picture Is Not Always What You Want, But That Is What We Give You** (Hassel and Sjöbergh 2005). On this Jonas Sjöbergh again did the main part of the implementation, myself and Jonas took equal part in the experiments, and Jonas had the main responsibility for writing the paper. All evaluation data collection for Swedish was performed by me.
- **Widening the HolSum Search Scope** (Hassel and Sjöbergh 2007). Here I did most of the implementation, performed two thirds of the experiments, as well as writing the main part of the paper.

5.2 Systems, Tools and Corpora

Conducting experiment-oriented research often requires the development of, besides data collection and evaluation methodologies, instruments for carrying out this research. These instruments may be bodies of data to be used for observation or learning, tools that permit you to collect or to make observations on said data, or systems that embody the result of such observations.

During the course of the research forming this thesis a set of applications, tools and corpora have been developed, refined or ported. These have to a high degree been necessary for the conducted research, and it is my strong belief that these will, and have been useful to others in their research. Therefore, these will here be given some attention here. For an overview of how the different resources are interconnected and employed, see Figure 5.1.[4]

SweSum

SweSum (Dalianis 2000) was first constructed by Hercules Dalianis in 1999, and has since been further developed and maintained by me. SweSum is a traditional extraction-based text summarizer, working on sentence level, which has HTML-tagged news text as its main domain. For each language the system utilizes a lexicon for mapping inflected forms of content words to their respective root. This is used for topic identification, based on the hypothesis that sentences containing

[4] To be explicit, the following resources have not been developed within the framework of this thesis, but have been used for training and evaluation: DUC 2001-2004 (provided by NIST), BNC (provided by the University of Oxford), SUC (provided by the department of linguistics at Stockholm University and Umeå University) and the Swedish Parole (provided Språkdata at Göterborg University).

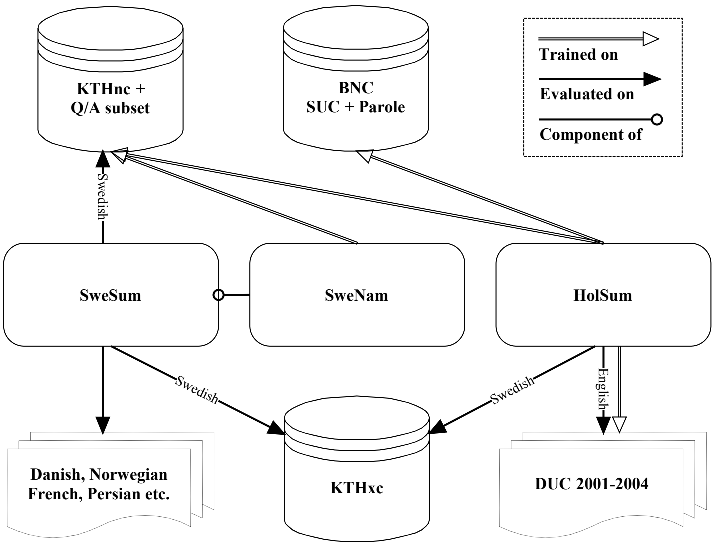


Figure 5.1: Overview of the main systems and corpora used, and their relationships.

high-frequent content words are central to the topic of the text. These observed frequencies are then modified by a set of heuristics, e.g. the sentence's position in the text, its formatting etc. Furthermore, SweSum requires an abbreviation lexicon and a set of language specific heuristics for correct tokenization and sentence splitting. SweSum has during its current life-span been ported to many languages[5] and has been extended with Unicode support.[6] For Swedish SweSum also sports rudimentary pronoun resolution (Hassel 2000) as well as named entity tagging (Dalianis and Åström 2001).

SweSum is freely available online at http://swesum.nada.kth.se.

HolSum

HolSum (Hassel and Sjöbergh 2005, 2006, 2007) is a summarizer that aims at providing overview summaries. It is trivial to port to different languages as it requires but a few very basic NLP tools. The system also presents a novel approach to accessing a summary, by calculating the summary's semantic similarity to the

[5] I would like to thank Dorte Haltrup (CST), Paul Meurer (UiB), Pascal Vaillant (ENST), Andrea Andrenucci & Marco Baroni (UniBo), Horacio Rodriguez (UPC) and Vangelis Karkaletsis & Stergos Afantenos (SKEL-NCSR) for their help with Danish, Norwegian, French, Italian, Spanish respectively Greek lexicons and feedback.

[6] I would also like to thank Nima Mazdak and Georgios Pachantouris for their adaptions to Farsi and Greek respectively.

text being summarized, before presenting it to the user. HolSum is thoroughly discussed in Section 3.5 and, in particular, in Paper 5.

JavaSDM

JavaSDM (Hassel 2006) is an open source Random Indexing (Sahlgren 2005) toolkit written in Java. It provides a full-fledged application for indexing as well as a small application for running some tests on semantic sets. An extensive set of parameters can be configured giving a high degree of control over the index being created. Also, all indexes are fully reproducible given the same data to operate on. The classes provided in the Java package are easy to integrate in a third party system on several levels, depending on what functionality is desired. JavaSDM is written and maintained by me, with kind help of master students and fellow colleagues.

KTH News Corpus

The KTH News Corpus (KTHnc) is a corpus of news articles, press releases and similar texts collected from the web during the course of about one and a half year. This corpus is further described and discussed in (Hassel 2001a). KTHnc was collected and the tools needed for data collection were developed and maintained by myself, and the main reason for discontinuing the collection of corpus data was due to the increasing labor updating the filters needed for removing HTML, JavaScript and other non-text elements from the retrieved text. The corpus contains approximately 13 million words (200,000 documents) to date.

KTH eXtract Corpus

The KTH eXtract Corpus (KTHxc) constitutes a web-based set of tools for collecting extractive summaries from informants as well as summarization evaluation (Hassel and Dalianis 2005). The corpus itself contains a number of original texts and several manual extracts for each text. The tool assists in the construction of the extract corpus by guiding the human informant creating a summary in such a way that only full extract units (most often sentences) are selected for inclusion in the summary. The interface allows for the reviewing of sentence selection at any time, as well as reviewing of the constructed summary before submitting it to the corpus. The tool-kit also provides useful statistics as inter-informant agreement, mean compression rate and variance. The actual corpus contains a small set of original texts, taken primarily from the news domain, in Swedish, Danish and English. For each text several extracts, approximately 20 extracts per text, have been provided by informants through the corpus collector interface.

KTH Q&A Corpus

The KTH Question and Answering Corpus (KTHqac) constitutes a subset of KTHnc containing 100 randomly selected texts. For each text a question central to the main topic of the text was formulated along with an answer to that question. This corpus, which was developed by myself, Hercules Dalianis and Ola Knutsson, was primarily developed for evaluating the impact of stemming in a Swedish search engine (Carlberger *et al.* 2001), but has since been used to evaluate the Swedish text summarizer SweSum (Dalianis and Hassel 2001).

5.3 Concluding Remarks

At the onset of this thesis research the aim was to bring summarization to the Scandinavian languages. During its course two tracks have somewhat in parallel been followed. The first track concerns using a generic but highly traditional sentence ranking architecture coupled with language specific lexica and handcrafted domain and language specific heuristics. This automatic text summarizer, SweSum, has mainly acted as a baseline system, but has also itself been actively evaluated during the development of the evaluation corpora. However, even given its generic architecture the porting of this system is not trivial.

The other track instead directed itself towards an even more generic architecture that in practice only requires rules for segmenting the text into meaning bearing constituents (e.g. words) and extraction units of suitable size (e.g. sentences). This is the track mainly treated in this thesis. This summarizer, HolSum, has proven to be easy to assemble using only a few very basic language processing tools. Although it does benefit from such operations as stemming and stopword filtering, as we have shown, at least the latter is easily implemented for many languages.

Several evaluation runs, of different traits, have also been conducted with differing degrees of human effort. The involvement of human informants or assessors seems inevitable in the evaluation of this kind of information management tool. However, the process of collecting gold-standard extraction-based corpora can to some extent be streamlined. Also the process of collecting large-scale corpora has proven to lend itself to a high degree of streamlining, thus making this process very lean on human resources. This means that one can build suitable language and domain specific corpora with relative ease and time, the latter of course by necessity given by the production rate within the desired language or domain.

An important aspect of the actual summarization method has proven to be the notion that summaries, on semantic grounds, can be compared content-wise to the original text, thereby making it possible to assess the fitness of the summary as an overview of the original text even before it is presented to the user. It should also be noted that this concept of self-evaluative summarization is not in any way constrained to extractive summarization, even though it is here used to differentiate between extractive summaries, nor is it to generated summaries even if they in this particular case are. The summaries being evaluated and selected from could in practice be produced by any means, even being man-made.

The Wide Blue Yonder

There is always more to want; more to investigate, to establish and to learn. There are always new and higher goals to reach, and of course there are still ideas connected to the research presented in this thesis that have not yet been proven, or for that matter, disproven. I will try to point out some of these loose ends below.

For instance, it would be interesting to investigate the holistic view on the summarization task further. For example, it would be interesting to get a clearer view of the highly uncharted vector space built by random indexing the source texts and summary candidates. One would most certainly want to sample a set of starting points for each source text in order to see how many end points this would result in. Obviously it would be nice if this resulted in only a few "best" summaries or, in the ideal case, only one. The latter is probably too much to hope for, but at least one would perhaps be able to establish to what extent starting in the leading sentences is the best strategy, and what tactics to adopt if this seems to fail. Perhaps this would result in a more adaptive model.

Another thing that would be worth investigating is how closely an ascent in cosine similarity between the semantic vector of the source text and the summary candidates also is mirrored in a similar ascent in e.g. ROGUE scores, compared to the starting point.

Furthermore, it would of course be very interesting to apply the "holistic" approach to more challenging languages, like Chinese which presents us with segmentation issues due to lacking word-boundary markers and the possible omission of punctuation marks, and Arabic which raises difficulties due to the fact that it is highly inflective and omits vowels in written form. These cases mainly pose difficulties in the preprocessing stage, but would in fact be a way to test the robustness of the approach.

An approach to largely language independent classification of different senses of the same lexical token (word) has been outlined in (Hassel 2005). It would be interesting to try this approach on a larger scale, and if promising, to investigate in what manner such derived semantic knowledge could be boot-strapped back into the semantic model used by HolSum.

Also, in order to further purify the property of language independence, it would be interesting to incorporate some approach to language independent stemming or lemmatization (Xu and Croft 1998, Bacchin *et al.* 2002, Dalianis and Jongejan 2006).

Lastly, but perhaps also primarily, the formation of document signatures by aggregating concept vectors, i.e. the constituting words' context vectors, demands further investigation. It is in serious need of solid validation unless a more linguistically based theory of the formation of semantic document signatures comes in demand.

Bibliography

Laura Alonso Alemany and Maria Fuentes Fort. 2003. Integrating Cohesion and Coherence for Automatic Summarization. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, April 12–17 2003.

ANSI. 1979. American National Standard for Writing Abstracts. Technical report, American National Standards Institute, Inc., New York, NY. ANSI Z39.14.1979.

Michela Bacchin, Nicola Ferro, and Massimo Melucci. 2002. The effectiveness of a graph-based algorithm for stemming. In *ICADL '02: Proceedings of the 5th International Conference on Asian Digital Libraries*, pages 117–128, London, UK. Springer-Verlag. ISBN 3-540-00261-8.

Breck Baldwin, Robert Donaway, Eduard Hovy, Elizabeth Liddy, Inderjeet Mani, Daniel Marcu, Kathleen McKeown, Vibhu Mittal, Marc Moens, Dragomir Radev, Karen Spärck-Jones, Beth Sundheim, Simone Teufel, Ralph Weischedel, and Michael White. 2000. An Evaluation Road Map for Summarization Research. http://www-nlpir.nist.gov/projects/duc/papers/summarization.roadmap.doc.

Harold Borko and Charles Bernier. 1975. *Abstracting Concepts and Methods.* Academic Press, New York.

Curt Burgess. 1998. From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model. *Behavior Research Methods, Instruments, & Company*, pages 189–198.

Lou Burnard. 1995. The Users Reference Guide for the British National Corpus.

N.G. Burton and Joseph Carl Robnett Licklider. 1955. Long-range constraints in the statistical structure of printed English. *American Journal of Psychology*, 68: 650–655.

Johan Carlberger, Hercules Dalianis, Martin Hassel, and Ola Knutsson. 2001. Improving Precision in Information Retrieval for Swedish using Stemming. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

Eugene Charniak. 1993. *Statistical Language Learning.* MIT Press, Cambridge, Massachusetts.

Edward T. Cremmins. 1996. *The Art of Abstracting*, 2nd edition. Information Resources Press, Arlington, VA.

David Crystal. 1987. *The Cambridge encyclopedia of language.* Cambridge University Press.

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical Report TRITA-NA-P0015, IPLab-174, KTH NADA, Sweden.

Hercules Dalianis and Erik Åström. 2001. SweNam - A Swedish Named Entity recognizer. Its construction, training and evaluation. Technical Report TRITA-NA-P0113, IPLab-189, KTH NADA, Sweden.

Hercules Dalianis and Martin Hassel. 2001. Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools. Technical Report TRITA-NA-P0112, IPLab-188, KTH NADA, Sweden.

Hercules Dalianis, Martin Hassel, Koenraad de Smedt, Anja Liseth, Till Christopher Lech, and Jürgen Wedekind. 2004. Porting and evaluation of automatic summarization. In H. Holmboe, editor, *Nordisk Sprogteknologi 2003: Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004.* Museum Tusculanums Forlag.

Hercules Dalianis, Martin Hassel, Jürgen Wedekind, Dorte Haltrup, Koenraad de Smedt, and Till Christopher Lech. 2003. Automatic Text Summarization for the Scandinavian Languages. In H. Holmboe, editor, *Nordisk Sprogteknologi 2002: Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004*, pages 153–163. Museum Tusculanums Forlag.

Hercules Dalianis and Bart Jongejan. 2006. Hand-crafted versus Machine-learned Inflectional Rules: The Euroling-SiteSeeker Stemmer and CST's Lemmatiser. In *Proceeding of the International Conference on Language Resources and Evaluation, LREC 2006*, Genoa, Italy.

Gaël de Chalendar, Romaric Besancon, Olivier Ferret, Gregory Grefenstette, and Olivier Mesnard. 2005. Thematic Extraction, Syntactic Sentence Simplification and Bilingual Generation towards Crosslingual Summarization. In *"Crossing Barriers in Text Summarization Research" workshop at RANLP'05*, Borovets, Bulgaria.

Koenraad de Smedt, Anja Liseth, Martin Hassel, and Hercules Dalianis. 2005. How short is good? An evaluation of automatic summarization. In H. Holmboe, editor, *Nordisk Sprogteknologi 2004: Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004.* Museum Tusculanums Forlag.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A Comparison of Rankings Produced by Summarization Evaluation Measures. In Udo Hahn, Chin-Yew Lin, Inderjeet Mani, and Dragomir R. Radev, editors, *Proceedings of the 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*, pages 69–78. Association for Computational Linguistics.

DUC. 2007. Document Understanding Conferences. http://duc.nist.gov/.

Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Scott C. Deerwester. 1988. Using latent semantic analysis to improve information retrieval. In *Proceedings of CHI'88: Conference on Human Factors in Computing*, pages 281–285, Washington, DC, USA, May 15-19 1988.

Harold Parkins Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Brigitte Endres-Niggemeyer, Elizabeth Maier, and Alexander Sigel. 1995. How to Implement a Naturalistic Model of Abstracting: Four Core Working Steps of an Expert Abstractor. *Information Processing & Management*, 31(5):631–674.

Thérèse Firmin and Michael J. Chrzanowski. 1999. An Evaluation of Automatic Text Summarization Systems. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 325–336. MIT Press.

George Forsythe, Michael Malcolm, and Cleve Moler. 1977. *Computer Methods for Mathematical Computations*. Prentice-Hall, Englewood Cliffs, NJ, USA.

William Gale, Kenneth Ward Church, and David Yarowsky. 1992. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, pages 249–256, Morristown, NJ, USA. Association for Computational Linguistics.

Yihong Gong and Xin Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, LA.

Gregory Grefenstette and Pasi Tapanainen. 1994. What is a word, what is a sentence? problems of tokenization. In *Proceedings of the 3rd International Conference on Computational Lexicography*, pages 79–87, Budapest, Hungary.

Leif Grönqvist. 2006. *Exploring Latent Semantic Vector Models Enriched With N-grams*. PhD thesis.

Donna Harman and Daniel Marcu, editors. 2001. *Proceedings of the 1st Document Understanding Conference*. New Orleans, LA.

Zelig S Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Martin Hassel. 2000. Pronominal Resolution in Automatic Text Summarisation. Master's thesis in computer science, Department of Computer and Systems Sciences (DSV), Stockholm University, Sweden.

Martin Hassel. 2001a. Internet as Corpus - Automatic Construction of a Swedish News Corpus. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

Martin Hassel. 2001b. newsAgent - A Tool for Automatic News Surveillance and Corpora Building. NUTEK report. http://www.nada.kth.se/∼xmartin/papers/Nutek.pdf.

Martin Hassel. 2003. Exploitation of Named Entities in Automatic Text Summarization for Swedish. In *Proceedings of NODALIDA'03 - 14th Nordic Conference on Computational Linguistics*, Reykjavik, Iceland, May 30-31 2003.

Martin Hassel. 2005. Word Sense Disambiguation Using Co-Occurrence Statistics on Random Labels. In *Proceedings of Recent Advances in Natural Language Processing 2005*, Borovets, Bulgaria.

Martin Hassel. 2006. JavaSDM - A Java tool-kit for working with Random Indexing. http://www.nada.kth.se/∼xmartin/java/JavaSDM/.

Martin Hassel and Hercules Dalianis. 2005. Generation of Reference Summaries. In *Proceedings of 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznan, Poland, April 21-23 2005.

Martin Hassel and Nima Mazdak. 2004. FarsiSum - A Persian Text Summarizer. In Ali Farghaly and Karine Megerdoomian, editors, *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages 82–84, Geneva, Switzerland, August 28th 2004. COLING.

Martin Hassel and Jonas Sjöbergh. 2005. A reflection of the whole picture is not always what you want, but that is what we give you. In *"Crossing Barriers in Text Summarization Research" workshop at RANLP'05*, Borovets, Bulgaria.

Martin Hassel and Jonas Sjöbergh. 2006. Towards holistic summarization: Selecting summaries, not sentences. In *Proceedings of LREC 2006*, Genoa, Italy.

Martin Hassel and Jonas Sjöbergh. 2007. Widening the HolSum Search Scope. In *Proceedings of NODALIDA'07 - 16th Nordic Conference on Computational Linguistics*, Tartu, Estonia, May 25-26 2007.

Eduard Hovy, editor. 1999. *Multilingual Information Management: Current Levels and Future Abilities. Chapter 3 Cross-lingual Information Extraction and Automated Text Summarization.* http://www.cs.cmu.edu/∼ref/mlim/chapter3.html.

Eduard Hovy and Chin-Yew Lin. 1997. Automated Text Summarization in SUMMARIST. In *Proceedings of the ACL97/EACL97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, July 1997.

Eduard Hovy, Chin-Yew Lin, and Liang Zhou. 2005. Evaluating duc 2005 using basic elements. Proceedings of Document Understanding Conference (DUC). Vancouver, B.C., Canada.

Eduard Hovy and Daniel Marcu. 1998. Automated Text Summarization Tutorial at COLING/ACL'98. http://www.isi.edu/∼marcu/acl-tutorial.ppt.

ISO 215:1986. 1986. Documentation – Presentation of Contributions to Periodicals and Other Serials. ISO 215:1986. Technical report, International Organisation for Standardisation.

Hongyan Jing. 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 310–315, Seattle,WA, April 29–May 4 2000.

Hongyan Jing and Kathleen R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In M. Hearst, Gey. F., and R. Tong, editors, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136, University of California, Beekely, August 1999.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and Paste-Based Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 178–185, Seattle, WA, April 2000.

Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random Indexing of text samples for Latent Semantic Analysis. In L.R. Gleitman and A.K. Josh, editors, *Proceedings 22nd Annual Conference of the Cognitive Science Society*, Pennsylvania, August 2000.

Ola Knutsson. 2001. *Automatisk språkgranskning av svensk text.* Licentiate thesis, KTH NADA, Sweden.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Thomas K. Landauer, Darrell Laham, Bob Rehder, and Missy E. Schreiner. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417, Stanford University, USA, August 7-10 1997.

Leah Larkey, Lisa Ballesteros, and Margaret Connell. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In *Proceedings of the 25th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 275–282, Tampere, Finland.

Alessandro Lenci, Roberto Bartolini, Nicoletta Calzolari, Ana Agua, Stephan Busemann, Emmanuel Cartier, Karine Chevreau, and JosÃ© Coch. 2002. Multilingual Summarization by Integrating Linguistic Resources in the MLIS-MUSI Project. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands, Spain, May 29-31 2002.

Elizabeth D. Liddy. 1991. Discourse-level Structure of Empirical Abstracts: An Exploratory Study. *Information Processing and Management*, 27(1):550–81.

Chin-Yew Lin. 2001. Summary Evaluation Environment (SEE). http://www.isi.edu/∼cyl/SEE.

Chin-Yew Lin. 2003. ROUGE: Recall-oriented understudy for gisting evaluation. http://www.isi.edu/∼cyl/ROUGE/.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th COLING Conference*, Saarbrücken, Germany.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Multi-Document Summarization Evaluation of the 2nd Document Understanding Conference at the 4Oth Meeting of the Association for Computational Linguistics*, Philadelphia, PA, July 2002.

Chin-Yew Lin and Eduard Hovy. 2003a. Automatic Evaluation of Summaries Using n-gram Co-occurrence Statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 27 - June 1 2003.

Chin-Yew Lin and Eduard Hovy. 2003b. The potential and limitations of automatic sentence extraction for summarization. In Dragomir Radev and Simone Teufel, editors, *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, Edmonton, Alberta, Canada, May 31 - June 1 2003. Association for Computational Linguistics.

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2(2):159–165.

Robert Wing Pong Luk. 1994. An ibm-pc environment for chinese corpus analysis. In *Proceedings of the 15th conference on Computational Linguistics*, pages 584–587.

Kevin Lund, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the Cognitive Science Society*, pages 660–665, Hillsdale, N.J.: Erlbaum Publishers.

Inderjeet Mani. 2001. Summarization Evaluation: An Overview. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*.

Inderjeet Mani, David House, G. Klein, Lynette Hirshman, Leo Orbst, Thérèse Firmin, Michael Chrzanowski, and Beth Sundheim. 1998. The TIPSTER SUMMAC Text Summarization Evaluation. Technical Report MTR 98W0000138, The Mitre Corporation, McLean, Virginia.

Inderjeet Mani and Mark T. Maybury, editors. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Daniel Marcu. 1999. The Automatic Construction of Large-Scale Corpora for Summarization Research. In M. Hearst, Gey. F., and R. Tong, editors, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–144, University of California, Berkely, August 1999.

John McCarthy. 1981. A Prosodic Theory of Nonconcatenative Morphology. *Linguistic Inquiry*, 12:373–418.

Rada Mihalcea. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 20, Barcelona, Spain.

Rada Mihalcea. 2005. Language Independent Extractive Summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, Ann Arbor, June 2005.

John Stuart Mill. 1843. *A System Of Logic, Raciocinative and Inductive.* London.

A. Morris, G. Kasper, and D. Adams. 1992. The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance. *Information Systems Research*, 3(1):17–35.

NIST. 2002. Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

M. Ortuño, P. Carpena, P. Bernaola-Galvan, E. Munoz, and A. Somoza. 2002. Keyword detection in natural languages and DNA. *Europhysics Letters*, 57:759–764.

Paul Over and James Yen. 2004. An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems. http://www-nlpir.nist.gov/projects/duc/pubs/2004slides/duc2004.intro.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A Method for Automatic Evaluation of Machine Translation. Research Report RC22176, IBM.

Prasad Pingali, Jagadeesh Jagarlamudi, and Vasudeva Varma. 2007. Experiments in Cross Language Query Focused Multi-Document Summarization. In *Proceedings of the Cross Lingual Information Access Addressing the Information Need of Multilingual Societies workshop at the International Joint Conference on Artificial Intelligence*, Hyderabad, India, January 6-12 2007.

Mirko Popovic and Peter Willett. 1992. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5):384–390.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies. In Udo Hahn, Chin-Yew Lin, Inderjeet Mani, and Dragomir R. Radev, editors, *Proceedings of the 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.

Dragomir R. Radev and Daniel Tam. 2003. Single-Document and Multi-Document Summary Evaluation via Relative Utility. In *Poster Session, Proceedings of the ACM CIKM Conference*, New Orleans, LA, November 2003.

Magnus Rosell. 2003. Improving clustering of Swedish newspaper articles using stemming and compound splitting. In *Proceedings of 14th Nordic Conference on Computational Linguistics – NODALIDA '03.*

Magnus Rosell and Sumithra Velupillai. 2005. The impact of phrases in document clustering for Swedish. In *Proceedings of 15th Nordic Conference on Computational Linguistics – NODALIDA '05.*

Horacio Saggion and Guy Lapalme. 2000. Concept Identification and Presentation in the Context of Technical Text Summarization. In Udo Hahn, Chin-Yew Lin, Inderjeet Mani, and Dragomir R. Radev, editors, *Proceedings of the 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, USA, April 30 2000. Association for Computational Linguistics.

Magnus Sahlgren. 2001. Vector-Based Semantic Analysis: Representing word meanings based on random labels. In *Proceedings of Semantic Knowledge Acquisition and Categorisation Workshop at ESSLLI'01*, Helsinki, Finland.

Magnus Sahlgren. 2002. Towards a Flexible Model of Word Meaning. In *Proceedings of the Workshop on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, AAAI Spring Symposium*, Stanford University, Palo Alto, California, USA, March 25-27 2002.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005)*, Copenhagen, Denmark, August 16 2005.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.* Doctoral thesis, Department of Linguistics, Stockholm University, Stockholm, Sweden.

Magnus Sahlgren, Preben Hansen, and Jussi Karlgren. 2002. English-Japanese Cross-lingual Query Expansion Using Random Indexing of Aligned Bilingual Text Data. In *Proceedings of the Third NTCIR Workshop*, Tokyo, Japan, October 8-10 2002.

Magnus Sahlgren and Jussi Karlgren. 2005. Automatic Bilingual Lexicon Acquisition Using Random Indexing of Parallel Corpora. *Journal of Natural Language Engineering, Special Issue on Parallel Texts*, 11(3).

Gerard Salton. 1971. *The SMART Retrieval System – Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Gerard Salton. 1988. *Automatic Text Processing.* Addison-Wesley Publishing Company.

Gerard Salton and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company.

Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic Text Structuring and Summarization. *Information Processing & Management*, 33(2):193–207.

Ferdinand de Saussure. 1916. *Course in General Linguistics (trans. Roy Harris, 1983).* Duckworth, London.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27.3-4:379–423,623–656.

Claude Elwood Shannon. 1951. Prediction and Entropy of Printed English. *The Bell System Technical Journal*, 30:50–64.

Advaith Siddharthan and Kathleen McKeown. 2005. Improving Multilingual Summarization: Using Redundancy in the Input to Correct MT errors. In *Proceedings of Human Language Technology / Empirical Methods in Natural Language Processing Conference (HLT/EMNLP 2005)*, Vancouver, Canada.

Jonas Sjöbergh. 2003. Bootstrapping a free part-of-speech lexicon using a proprietary corpus. In *Proceedings of ICON-2003: International Conference on Natural Language Processing*, pages 1–8, Mysore, India.

Jonas Sjöbergh. 2005. Chunking: an unsupervised method to find errors in text. In *Proceedings of NODALIDA'05 - 15th Nordic Conference on Computational Linguistics*, Joensuu, Finland.

Jonas Sjöbergh. 2007. Older versions of the ROUGEeval summarization evaluation system were easier to fool. *Journal of Information Processing and Management, Special Issue on Summarization*, doi:10.1016/j.ipm.2007.01.014.

Jonas Sjöbergh and Viggo Kann. 2004. Finding the correct interpretation of Swedish compounds a statistical approach. In *Proceedings of LREC-2004*, pages 899–902, Lisbon, Portugal.

Karen Spärck-Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation)*, 28:11–21.

Karen Spärck-Jones. 1999. Automatic Summarizing: Factors and Directions. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–13. The MIT Press.

Karen Spärck-Jones and Julia R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review.* Number 1083 in Lecture Notes in Artificial Intelligence. Springer.

Cornelis Joost Van Rijsbergen. 1979. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow. http://www.dcs.gla.ac.uk/Keith/Preface.html.

Adam Winkel and Dragomir Radev. 2002. MEADeval: An evaluation framework for extractive summarization. http://perun.si.umich.edu/clair/meadeval/.

Jinxi Xu and Bruce Croft. 1998. Corpus-based Stemming using Co-occurrence of Word Variants. *ACM Transactions on Information Systems*, 16(1):61–81.

Jen-Yuan Yeh, Hao-Ren Ke, and Wei-Pang Yang. 2002. Chinese text summarization using a trainable summarizer and latent semantic analysis. In *ICADL '02: Proceedings of the 5th International Conference on Asian Digital Libraries*, pages 76–87, Singapore.

Included Papers

Paper 1.
Internet as Corpus – Automatic Construction of a Swedish News Corpus (Hassel 2001a)

Paper 2.
Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools (Dalianis and Hassel 2001)

Paper 3.
Exploitation of Named Entities in Automatic Text Summarization for Swedish (Hassel 2003)

Paper 4.
Generation of Reference Summaries (Hassel and Dalianis 2005)

Paper 5.
Navigating Through Summary Space – Selecting Summaries, Not Sentences (Hassel and Sjöbergh, submitted 2007)

Paper 1

Internet as Corpus – Automatic Construction of a Swedish News Corpus

Martin Hassel. 2001. *Internet as Corpus – Automatic Construction of a Swedish News Corpus.* In Proceedings of NODALIDA'01 – 13th Nordic Conference on Computational Linguistics, Uppsala, Sweden, May 21–22 2001.

Internet as Corpus – Automatic Construction of a Swedish News Corpus

Martin Hassel
KTH NADA
Royal Institute of Technology
100 44 Stockholm, Sweden
xmartin@nada.kth.se

Abstract

This paper describes the automatic building of a corpus of short Swedish news texts from the Internet, its application and possible future use. The corpus is aimed at research on Information Retrieval, Information Extraction, Named Entity Recognition and Multi Text Summarization. The corpus has been constructed using an Internet agent, the so called *newsAgent*, downloading Swedish news text from various sources. A small part of this corpus has then been manually tagged with keywords and named entities. The newsAgent is also used as a workbench for processing the abundant flow of news texts for various users in a customized format in the application *Nyhetsguiden*.

1 Introduction

Two years ago we built an automatic text summarizer called SweSum for Swedish text (Dalianis 2000). We wanted to evaluate SweSum but there were no tagged Swedish corpora available to evaluate text summarizers or information retrieval tools processing Swedish as there are for the English speaking community, mainly through the TREC (Vorhees and Tice 2000), MUC and TIPSTER-SUMMAC evaluation conferences (Mani *et al.* 1998, Krenn and Samuelsson 1997). The purpose of this project[1] was to construct a test bed for new natural language technology tools, i.e. *automatic text summarization, named entity tagging, stemming, information retrieval/extraction* etc. In the process of building this system, Nyhetsguiden (Hassel 2001), we also made it capable of gathering the news texts into a corpus, a

[1] This project is supported by NUTEK (Swedish board for Industrial and Technical Development) FavorIT programme in cooporation with EuroSeek AB.

corpus we have used to train and evaluate such tools as mentioned above. As this corpus is aimed at research on information and language technology applied on redundant text, the system does not, contrary to (Hofland 2000), remove duplicated concordance lines.

2 Nyhetsguiden – A User Centred News Delivery System

The system has a modular design and consists of three parts, the user interface, the user database and the main application, newsAgent. Being modular, the system can be run as a distributed system or on a single web server. When run as a distributed system, at least newsAgent must be run on a computer with Internet access. The user interface (Nyhetsguiden) and the user database can reside on either an Internet or Intranet capable server depending on the desired public access to the system. newsAgent is the core of the system and is basically a web spider that is run in a console window. The spider is implemented in Perl, which makes it platform independent, that is, it can run on any platform running Perl (Unix/Linux, Windows, Macintosh, BeOS, Amiga, etc). On intervals of 3–5 minutes newsAgent searches the designated news sources (see Appendix A) for new news texts, that is news texts not seen by the system before. When a new news text is encountered it is fetched, the actual news text and accompanying illustrations are extracted (by removing navigation panels, banners, tables of links, etc). The resulting document is then passed through the system and, depending on configuration; stored, summarized and routed to the end recipient.

3 Construction of a Corpus of Swedish News Texts

Traditionally it has been hard work constructing a corpus of news text. In Sweden there are no newspapers that on a yearly basis offer their paper in digital form,[2] as some foreign newspapers do (for example Wall Street Journal), meaning that obtaining this material has to be done on demand. Many Swedish newspapers are, when inquired, unwilling to release texts from their archives for research purposes, and even when they do, it is often the question of a small amount of news texts with an age of several years. This may potentially lead to the exclusion of contemporary words and giving unusually high, or low, occurrence frequencies to words related to phenomena limited to a certain period of time.

[2] We have as yet only been able to aquire 1995 years issue of Svenska Dagbladet (SvD), also the Scarrie Swedish News Corpus (Dahlqvist 1998) contains all articles published in SvD and Uppsala Nya Tidning (UNT) during the same period.

In the past the solution would have been to collect newspapers in their paper form and type or scan them, using an Optical Character Recognition program, in order to convert them to a format manageable by computers.

The World Wide Web is, on the other hand, today a large collection of texts written in different languages and thus giving an abundant resource for language studies already in a format, by necessity, manageable by computers. Many of the web pages are also frequently updated and thus give us a steady access to concurrent use of language in different fields. In this situation, neglecting the usability of Internet as a corpus would be foolish. In our case we used a tool called newsAgent that is a set of Perl scripts designed for gathering news texts, news articles and press releases from the web and routing them by mail according to subscribers' defined information needs.

4 KTH News Corpus

The project with the KTH News Corpus was initiated in May 2000. We started out collecting news telegrams, articles and press releases from three sources but with the ease of adding new sources we settled for twelve steady news sources (Appendix A). The choice of these news sources was based partly on site and page layout, partly on the wish to somewhat balance the corpus over several types of news topics. Among the chosen news sources are both general news, "daily press", and specialized news sources. The reason for this is the possibility of comparing how the same event is described depending on targeted reader (wording, level of detail etc). As of February 2001 we have gathered more than 100,000 texts amounting to over 200 Mb with an increase of over 10,000 new texts each month. The increase in word forms during March was almost 230,000. The lengths of the texts vary between 5 and 500 sentences with a tendency towards the shorter and an average length of 193 words per text.

The texts are stored in a HTML tagged format but only the news heading and the body of the news text is preserved. All other page layout and all navigation tables and banners are removed. Each text is tagged with meta tags storing the information on time and date of publication, news source and source URL. We stored the news in different categories, see Appendix A, thus giving us the possibility to study the difference in use of language in, for example, news on cultural respectively sports event. We did this using the news sources own categorization of their news texts (finance, sports, domestic, foreign etc), instead of a reader based categorization, such as described in (Karlgren 2000). The corpus is structured into these categories by the use of catalogue structure, a Hypertext linked index and a search engine driven index thus giving several modes of orientation in the corpus.

For the purpose of evaluating a Swedish stemmer in conjunction with a search engine, we manually tagged 100 texts TREC style and constructed questions and answers central to each text (Carlberger *et al.* 2001). We also tagged each text with named entities (names, places, organisations and date/time) and the five most significant keywords for future evaluation purposes.

Unfortunately copyright issues remain unsolved, we have no permission from the copyright holders except fair use, and so the corpus can only be used for research within our research group. The tool for gathering the corpus, newsAgent, is on the other hand available for use outside our research group (with the exclusion of mail routing and FTP plug-ins).

4.1 Areas of Use

So far the corpus has been used for evaluation and training purposes. Knutsson (2001) has employed the corpus for evaluating error detection rules for Granska, a program for checking for grammatical errors in Swedish unrestricted text (Domeij *et al.* 1999). The tagged texts have besides being used for evaluation of a Swedish stemmer (Carlberger *et al.* 2001) also been utilized in the evaluation of SweSum, an automatic text summarizer that among other languages handles Swedish unrestricted HTML tagged or untagged ASCII text (Dalianis and Hassel 2001) and for the training and evaluation of a Named Entity Tagger, SweNam (Dalianis and Åström 2001).

In the near future parts of the corpus will be used and for expanding SweSum with Multi Text Summarization. Other possible areas of use are for producing statistics and lexicons, and for developing a Topic Detection Tracking (for example, see Wayne 2000) system for Swedish news. This will hopefully result in a tool that in a short period can build a corpus of plain, tagged and summarized versions of the same news text along with appropriate statistics.

5 Conclusions

A concluding remark is that a small piece of programming has grown to a complete system which we had great use of in training and evaluation of various natural language tools and that the newsAgent has been a incentive to push our research beyond foreseeable limits. As a part of our online service Nyhetsguiden we have also gained as much as fifty willing beta testers of our language technology tools. We are now on the verge to incorporate our new Named Entity Tagger into newsAgent. We also believe that this proves that it is feasible to acquire a substantial corpus, over a short period of time, from the Internet. One may argue that as long as copyright issues are not solved, the corpus has no legal use outside our research group. While this is true, the corpus has been of great use to us in our research and the corpus tools still remain free for public use. The tools have proven to be practically service free and run without major problems. Since the same news reports are, potentially, repeated over news sources and time, the resulting corpus will be of much use for research on Information Extraction/Retrieval and Topic Detection Tracking.

References

Johan Carlberger, Hercules Dalianis, Martin Hassel, and Ola Knutsson. 2001. Improving Precision in Information Retrieval for Swedish using Stemming. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

B. Dahlqvist. 1998. The SCARRIE Swedish News Corpus. In Anna Sgwall Hein, editor, *eports from the SCARRIE project*. Uppsala University.

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical Report TRITA-NA-P0015, IPLab-174, KTH NADA, Sweden.

Hercules Dalianis and Erik Åström. 2001. SweNam - A Swedish Named Entity recognizer. Its construction, training and evaluation. Technical Report TRITA-NA-P0113, IPLab-189, KTH NADA, Sweden.

Hercules Dalianis and Martin Hassel. 2001. Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools. Technical Report TRITA-NA-P0112, IPLab-188, KTH NADA, Sweden.

Rickard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. 1999. Granska - An efficient hybrid system for Swedish grammar checking. In *Proceedings of NODALIDA'99 - 12th Nordic Conference on Computational Linguistics*.

Martin Hassel. 2001. newsAgent - A Tool for Automatic News Surveillance and Corpora Building. NUTEK report. http://www.nada.kth.se/∼xmartin/papers/Nutek.pdf.

K. Hofland. 2000. A self-expanding corpus based on newspapers on the Web. In *In Proceedings of Second Internation Conference on Language Resources and Evaluation. LREC-2000*, Athens, Greece, May 31 - June 2 2000.

Jussi Karlgren. 2000. *Assembling a Balanced Corpus from the Internet. In Stylistic Experiments for Information Retrieval.* PhD thesis, Department of Linguistics, Stockholm University, Sweden.

Ola Knutsson. 2001. *Automatisk språkgranskning av svensk text*. Licentiate thesis, KTH NADA, Sweden.

Brigitte Krenn and Christer Samuelsson, editors. 1997. *The Linguist's Guide to Statistics - Don't Panic*. http://www.coli.uni-sb.de/∼krenn/edu.html.

Inderjeet Mani, David House, G. Klein, Lynette Hirshman, Leo Orbst, Thérèse Firmin, Michael Chrzanowski, and Beth Sundheim. 1998. The TIPSTER SUMMAC Text Summarization Evaluation. Technical Report MTR 98W0000138, The Mitre Corporation, McLean, Virginia.

E.M. Vorhees and D.M. Tice. 2000. The TREC-8 Question Answering System Track. In *In the proceedings of Second International Conference on Language Resources and Evaluation. LREC-2000*, Athens, Greece, May 31 - June 2 2000.

C. Wayne. 2000. Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation. In *In the proceedings of Second International Conference on Language Resources and Evaluation. LREC-2000*, Athens, Greece, May 31 - June 2 2000.

Appendix A

News sources and categories used by newsAgent:

Aftonbladet	- Economics, cultural, sports, domestic & foreign news
Amnesty International	- Press releases and news on human rights
BIT.se (Sifo Group)	- Press releases from companies
Dagens Industri	- News on the industrial market
Dagens Nyheter	- Economics, cultural, sports, domestic & foreign news
Homoplaneten (RFSL)	- News concerning rights of the homosexual community
Tidningen Mobil	- News articles on mobile communication
International Data Group	- News articles on computers
Medstrms Frlag	- News articles on computers
Senaste Nytt.com	- News flashes (discontinued)
Svenska Dagbladet	- News flashes
Svenska Eko-nyheter	- News flashes
Sveriges Riksdag	- Press releases from the Swedish Parliament

Paper 2

Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools

Hercules Dalianis and Martin Hassel. 2001. *Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools.* Technical report, KTH NADA, Sweden.

Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools

Hercules Dalianis and Martin Hassel
KTH NADA
Royal Institute of Technology
100 44 Stockholm, Sweden
{hercules,xmartin}@nada.kth.se

Abstract

We are presenting the construction of a Swedish corpus aimed at research[1] on Information Retrieval, Information Extraction, Named Entity Recognition and Multi Text Summarization, we will also present the results on evaluating our Swedish text summarizer SweSum with this corpus. The corpus has been constructed by using Internet agents downloading Swedish newspaper text from various sources. A small part of this corpus has then been manually annotated. To evaluate our text summarizer SweSum we let ten students execute our text summarizer with increasing compression rates on the 100 manually annotated texts to find answers to predefined questions. The results showed that at 40 percent summarization/compression rate the correct answer rate was 84 percent.

1 Introduction

Two years ago we built a text summarizer called SweSum[2] (Dalianis 2000) for Swedish text. We wanted to evaluate SweSum but there were no annotated Swedish corpora available to evaluate text summarizers or information retrieval tools processing Swedish as there are for the English speaking community, mainly through the TREC (Vorhees and Tice 2000), MUC and TIPSTER-SUMMAC evaluation conferences (Mani *et al.* 1998, Krenn and Samuelsson 1997).

[1] This project is supported by NUTEK (Swedish board for Industrial and Technical Development) FavorIT programme in cooperation with Euroseek AB.

[2] SweSum is available online for testing at http://swesum.nada.kth.se and is also available for Norwegian, Danish, English, Spanish, French and German.

The only annotated corpora so far for Swedish is the Stockholm-UmeåSUC (1 million words, manually morpho-syntactically annotated) balanced corpus for evaluation of taggers (Ejerhed *et al.* 1992) and the Swedish Parole corpus aimed at language studies (Språkdata 2000). The text material in the Parole corpus is morpho-syntactically tagged with a statistical tagger. The corpus is balanced, contains approximately 18.5 million words and is available from Språkdata, which is affiliated with Göteborgs Universitet.

One interesting approach to create an evaluation corpus for Swedish is a technique described by Marcu (1999). This technique requires a text and its abstract. From these two inparameters one can create an extract automatically which can be used to assess a text summarizer, but we had no Swedish texts with abstracts available.

Lacking the appropriate tools we managed to make a subjective evaluation of SweSum using the techniques described by Firmin and Chrzanowski (1999). They write that one can make qualitative, subjective, intrinsic evaluations of the text by investigating if the text is perceived as well formed in terms of coherence and content. Therefore we let a number of students within the framework of 2D1418 Språkteknologi (Human Language Technology), a 4-credit course at NADA/KTH, Stockholm, in the fall 1999, automatically summarize an identical set of ten texts each of news articles and movie reviews using our text summarizer SweSum. The purpose was to see how much a text could be summarized without loosing coherence or important information. We found that the coherence of the text was intact at 30 percent compression rate and that the information content was intact at 25 percent compression rate (Dalianis 2000).[3] But to make an objective evaluation we needed an annotated corpus or at least a partly annotated corpus.

The only way to make this possible was to construct a Swedish annotated corpus ourselves. We also needed such an annotated corpus to evaluate our Swedish stemming algorithm; see (Carlberger *et al.* 2001). This was two of the reasons to create a Swedish corpus for evaluation of IR-tools.

2 Constructing the Corpus

Traditionally it has been hard work constructing a corpus of news text. In Sweden there are no newspapers that on a yearly basis offer their paper in digital form, as some foreign newspapers do. This means that obtaining news texts has to be done on demand. Many Swedish newspapers are, when inquired, unwilling to release texts from their archives for research purposes, and even when they do, it is often

[3] The compression rate is defined as the number of words in the summary text divided by number of words in the source text

the question of a small amount of news texts with an age of several years. This may potentially lead to the exclusion of contemporary words and giving unusually high, or low, occurrence frequencies to words related to phenomena limited to a certain period of time.

In the past the solution would be to collect newspapers in their paper form and type or scan them, using a Optical Character Recognition program, in order to convert them to a format manageable by computers.

The World Wide Web is, on the other hand, today a large collection of texts written in different languages and thus is an abundant resource for language studies already in a format, by necessity, manageable by computers. Many of the web pages are also frequently updated and so give a steady access to concurrent use of language in different fields. In this situation, neglecting the usability of Internet as a corpus would be irrational. In our case we used a tool called newsAgent that is a set of Perl programs designed for gathering news articles and press releases from the web and routing them by mail according to subscribers defined information needs.

3 Downloading and Storing

The project with the KTH News Corpus was initiated in May 2000. We started out automatically collecting news telegrams, articles and press releases in Swedish from three sources but with the ease of adding new sources we soon settled for twelve steady news sources.

The choice of these news sources was based partly on site and page layout, partly on the wish to somewhat balance the corpus over several types of news topics. Among the chosen news sources are both general news, “daily press”, and specialized news sources. The reason for this is the possibility of comparing how the same event is described depending on targeted reader (wording, level of detail etc).

As of February 2001 we have gathered more than 100,000 texts amounting to over 200 Mb with an increase of over 10,000 new texts each month. The increase in word forms during March was almost 230,000. The lengths of the texts vary between 5 and 500 lines with a tendency towards the shorter and an average length of 193 words per text.

The texts are stored in a HTML tagged format but only the news heading and the body of the news text is preserved. All other page layout and all navigation tables and banners are removed. Each text is tagged with meta tags storing the information on time and date of publication, news source and source URL. Using the news sources own categorization of their news texts, instead of a reader based categorization (Karlgren 2000), we have stored the news in different categories. This gives the possibility to study the difference in use of language in, for example, news on cultural respectively sports events. The corpus is structured into these categories by the use of catalogue structure, a HyperText linked index and a search engine driven index thus giving several modes of orientation in the corpus.

Since the purpose of the corpus is research on Information Retrieval, Information Extraction, Named Entity Recognition and Multi Text Summarization the system does not, contrary to the approach presented by Hofland (2000), remove duplicated concordance lines.

4 Annotation

From the downloaded corpus we selected 54,487 news articles from the period May 25, 2000 to November 4, 2000 and from these text we decided to manually annotate 100 news articles.

Three different persons constructed the Question and Answering (Q&A) schema, in total 100 questions and answers, (33, 33 and 34 Q&A respectively each), by randomly choosing among the 54 487 news articles from the KTH News corpus, for each constructing a question from the text, finding the answer in the text and annotating the text with: Filename, Person, Location, Organization, Time and five keywords. The 100 texts had an average length of 181 words each.

The reason to have the above tag-set was that the corpus is used and will be used to many tasks, namely, evaluation of an IR tool, (Carlberger *et al.* 2001), Text Summarization, Multi Text Summarization, Name Entity (NE) recognition and key word extraction. We constructed a Question and Answering annotation schema, see Figure 1, following the annotation standard in (Mani *et al.* 1998).

5 Evaluation

Objective methods to evaluate text summarizers are described in (Mani *et al.* 1998), one of these methods is to compare the produced summary (mainly extracts) with manually made extracts from the text to judge the overlap and consequently assess the quality of the summary. One other objective method to evaluate text summarizers is taken from the information retrieval area where a Question and Answering schema is used to reveal if the produced summary is the “right one”.

A text summarizer summarizes a text and a human assesses if the summary contains the answer of a given question. If the answer is in the summarized text then the summary is considered valid.

```
<top>
<num> Number: 35
<desc> Description: (Natural Language question)
    Vem ar koncernchef pa Telenor? (Who is CEO at Telenor?)
</top>

<top>
<num> Number: 35
<answer> Answer: Tormod Hermansen
<file> File: KTH NewsCorpus/Aftonbladet/Ekonomi/8621340_EKO__00.html
<person> Person: Tormod Hermansen
<location> Location: Norden
<organization> Organization: Telenor
<time> Time: onsdagen
<keywords> Keywords: Telenor; koncernchef; teleforetag; uppkop
</top>
```

Figure 1: Question and Answer tagging scheme.

We let ten students within the framework of 2D1418 Språkteknologi (Human Language Technology), a 4-credit course at NADA/KTH, Stockholm, in the fall 2000, automatically summarize a set of ten news articles each using the text summarizer SweSum at increasing compression rates; 20, 30 and 40 percent. If the 20, 30 and 40 percent summaries failed then the users could select their own keywords to direct the summarizer at 20 percent compression rate to find the answers to the predefined questions. We then compared the given answers with the correct ones. The results are listed in Table 1 below.

Summary/ Compresssion rate	**20%**	**30%**	**40%**	**Keywords at 20%**	**Correct answers**
Number of texts	97	97	97	97	
Given and correct answers	50	16	15	4	85
Percent accumulated correct answers	52%	68%	84%	88%	

Table 1. Evaluation of the text summarizer SweSum.

From the evaluation at 20 percent compression rate we can conclude that we obtained 52 percent correct answers and at 40 percent compression rate we obtained totally 84 percent correct answers, only 12 summaries did not give any answer at all (some of the them did not become summarized due to technical problems).

We noted during the annotation phase that if we had constructed questions with a yes answer or a one-word answer instead of a long ambiguous complicated answer then we could had automated the evaluation process since the computer automatically could check if the manually given answer is correct or not.

6 Conclusions

We have constructed the first Swedish corpus for evaluating text summarizers and similar information retrieval tools. We found that our text summarizer SweSum at 40 percent compression rate gave 84 percent correct answers. From this evaluation we can conclude that our summarizer for Swedish is state-of-the-art compared to other summarizers for English (Mani *et al.* 1998). Comparing our current objective evaluation results we can also validate that our previous subjective evaluation results (Dalianis 2000) were correct, indicating that a 30 percent compression rate gave good summaries.

There is no perfect summarization, every person has his preference when creating an abstract from a text. Except for the evaluation of the text summarizer SweSum, the corpus has been used for three other evaluation purposes: First, for evaluating our Swedish stemming algorithm (Carlberger *et al.* 2001) (we obtained 15 percent improvement in precision and 18 percent improvement on relative recall using stemming for Swedish); second, for evaluating our Swedish Named Entity recognizer - SweNam (Dalianis and Åström 2001) (we obtained 92 percent precision and 46 percent recall); and third, for evaluating error detection rules for Granska, a program for checking for grammatical errors in Swedish unrestricted text (Knutsson 2001).

Unfortunately copyright issues remain unsolved so the corpus can only be used for research within our research group. The tool for gathering the corpus, newsAgent, is on the other hand freely available for use outside our research group (with the exclusion of mail routing and FTP plug-ins).

References

Johan Carlberger, Hercules Dalianis, Martin Hassel, and Ola Knutsson. 2001. Improving Precision in Information Retrieval for Swedish using Stemming. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical Report TRITA-NA-P0015, IPLab-174, KTH NADA, Sweden.

Hercules Dalianis and Erik Åström. 2001. SweNam - A Swedish Named Entity recognizer. Its construction, training and evaluation. Technical Report TRITA-NA-P0113, IPLab-189, KTH NADA, Sweden.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. *SUC - The Stockholm-Umeå Corpus*, version 1.0 (suc 1.0). CD-ROM produced by the Dept of Linguistics, University of Stockholm and the Dept of Linguistics, University of Umeå. ISBN 91-7191-348-3.

Thérèse Firmin and Michael J. Chrzanowski. 1999. An Evaluation of Automatic Text Summarization Systems. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*, pages 325–336. MIT Press.

K. Hofland. 2000. A self-expanding corpus based on newspapers on the Web. In *In Proceedings of Second Internation Conference on Language Resources and Evaluation. LREC-2000*, Athens, Greece, May 31 - June 2 2000.

Jussi Karlgren. 2000. *Assembling a Balanced Corpus from the Internet. In Stylistic Experiments for Information Retrieval.* PhD thesis, Department of Linguistics, Stockholm University, Sweden.

Ola Knutsson. 2001. *Automatisk språkgranskning av svensk text.* Licentiate thesis, KTH NADA, Sweden.

Brigitte Krenn and Christer Samuelsson, editors. 1997. *The Linguist's Guide to Statistics - Don't Panic.* http://www.coli.uni-sb.de/∼krenn/edu.html.

Inderjeet Mani, David House, G. Klein, Lynette Hirshman, Leo Orbst, Thérèse Firmin, Michael Chrzanowski, and Beth Sundheim. 1998. The TIPSTER SUMMAC Text Summarization Evaluation. Technical Report MTR 98W0000138, The Mitre Corporation, McLean, Virginia.

Daniel Marcu. 1999. The Automatic Construction of Large-Scale Corpora for Summarization Research. In M. Hearst, Gey. F., and R. Tong, editors, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–144, University of California, Berkely, August 1999.

Språkdata. 2000. The Swedish PAROLE Lexicon.
http://spraakdata.gu.se/parole/lexikon/swedish.parole.lexikon.html.

E.M. Vorhees and D.M. Tice. 2000. The TREC-8 Question Answering System Track. In *In the proceedings of Second International Conference on Language Resources and Evaluation. LREC-2000*, Athens, Greece, May 31 - June 2 2000.

Paper 3

Exploitation of Named Entities in Automatic Text Summarization for Swedish

Martin Hassel. 2003. *Exploitation of Named Entities in Automatic Text Summarization for Swedish.* In Proceedings of NODALIDA'03 – 14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland, May 30–31 2003.

Exploitation of Named Entities in Automatic Text Summarization for Swedish

Martin Hassel
KTH NADA
Royal Institute of Technology
100 44 Stockholm, Sweden
xmartin@nada.kth.se

Abstract

Named Entities are often seen as important cues to the topic of a text. They are among the most information dense tokens of the text and largely define the domain of the text. Therefore, Named Entity Recognition should greatly enhance the identification of important text segments when used by an (extraction based) automatic text summarizer. We have compared Gold Standard summaries produced by majority votes over a number of manually created extracts with extracts created with our extraction based summarization system, SweSum. Furthermore we have taken an in-depth look at how overweighting of named entities affects the resulting summary and come to the conclusion that weighting of named entities should be carefully considered when used in a naïve fashion.

1 Background

The technique of automatic text summarization has been developed for many years (Luhn 1958, Edmundson 1969, Salton 1988). One way to do text summarization is by text extraction, which means to extract pieces of an original text on a statistical basis or with heuristic methods and put them together to a new shorter text with as much information as possible preserved (Mani and Maybury 1999).

One important task in text extraction is topic identification. There are many methods to perform topic identification (Hovy and Lin 1997). One is word counting at concept level, which is more advanced than just simple word counting; another is identification of cue phrases to find the topic.

To improve our automatic text summarizer and to a larger extent capture the topic of the text we tried to use Named Entity Recognition. Named Entity recognition is the task of finding and classifying proper nouns in running text. Proper nouns, such as names of persons and places, are often central in news reports. Therefore we have integrated a Named Entity tagger with our existing summarizer, SweSum, in order to study its effect on the resulting summaries.

2 Introducing SweSum

The domain of SweSum (Dalianis 2000) is Swedish newspaper text. SweSum utilizes several different topic identification schemes. For example the bold tag is often used to emphasize contents of the text. Headings are also given a higher weight. In news paper text the most relevant information is most often presented at the top. In some cases the articles are even written to be cuttable from from the bottom. Because of this we use Position Score (Hovy and Lin 1997); sentences in the beginning of the text are given higher scores than later ones.

Sentences that contain keywords are scored high. A keyword is an open class word with a high Term Frequency (*tf*). Sentences containing numerical data are also considered carrying important information. All the above parameters are put into a naïve combination function with modifiable weights to obtain the total score of each sentence.

3 Working Hypothesis

Named entities are often seen as important cues to the topic of a text. They are among the most information dense tokens of the text and largely define the domain of the text. Therefore, Named Entity Recognition should greatly enhance the identification of important text segments when used by an (extraction based) automatic text summarizer.

4 Enter SweNam

For Named Entity recognition and classifying SweNam (Dalianis and Åström 2001) is used. SweNam acts as a preprocessor for SweSum and tags all found named entities with one of the four possible categories - names of persons (given name and/or surname), locations (geographical as well as geopolitical), companies (names of companies, brands, products, organizations, etc) and time stamps (dates, weekdays, months etc). The named entities found by SweNam are quite reliable, as it has shown a precision of 92 percent. However, the recall is as low as 46 percent, so far from all named entities are considered during the summarization phase.

All found entities are given an equal weight and entered, together with the parameters described above, into the combination function in weighting module in the summarizer, SweSum.

5 Creating a Gold Standard

For the evaluation we collected two sets of texts, each set consisting of 10 news texts. The first set (Group 1) consisted of ten news articles randomly chosen from Svenska Dagbladet's web edition (http://www.svd.se/) over a couple of days. These where summarized using SweSum both with and without the use of Named Entity Recognition.

In order to evaluate and compare the two subsets of generated extracts from Group 1 we devised a system to collect manual extracts for the news articles from Group 1. Human test subjects where presented with the news articles one at a time in random order in the form of one sentence per line. In front of each line was a checkbox with which the informant could select that particular sentence for extraction. The informant could then choose to generate an extract based on the selected sentences. This extract was then presented to the informant who had to accept the extract before it was entered into a database. Submitted extracts were allowed to vary between 5% and 60% of the original text length.

The result was that 11 informants submitted a total of 96 extracts for the ten texts of Group 1. Each news text received between 8 and 11 manual extracts and the mean length of submitted extracts was 37%. These manual extract constituted the foundation for the KTH eXtract Corpus.

There was, as expected, not very much agreement between the informants on which sentences to select for the extract. The level of agreement among the informants was calculated with a simple precision function. This is done per text and then a mean was calculated over all ten texts.

$$AgreementLevel = \frac{Vc}{Ns \times Nx} \times 100$$

In the function above *Vc* is the number of votes that are represented in the generated extract, *Ns* is the number of sentences represented in the same extract and *Nx* is the number of man-made extracts made for the original text the votes and sentences account for. This means that when all informants choose not only the same number of sentences but also exactly the same set of sentences the function will result in a precision, or agreement, of 100%.

We where prepared for a low agreement among the human extractors as to which sentences are good summary sentences as previous studies have shown this (for an overview see Mani (2001). When taking all selected extraction units into account for each text there was only a mean agreement of 39.6%. This is however not so bad as it can seem at first glance. When generating a “gold standard” extract by presenting the most selected sentences up to a summary length of the mean length of all man-made extracts for a given text the precision, or the agreement level, rose to 68.9%. Very few of the sentences chosen for the gold standard where selected by as few as one third or less of the informants. Of course, even fewer sentences where selected by all informants. In fact, not even all informants could agree upon extracting the title or not when one was present in the source text.

6 Evaluation

The extract summaries generated with SweSum where then manually compared on sentence level with the gold standard summaries generated by majority vote. We found that with Named Entity Recognition the summaries generated by SweSum and the gold standard only had 33.9% of their sentences in common (see Table 1). On the other hand, without Named Entity Recognition the summaries generated with SweSum shared as many as 57.2% of the sentences with the gold standard.

	With NER	**Without NER**
Shared sentences	33.9%	57.2%

Table 1. Gold standard compared to SweSum generated extracts.

Of course this does not say much about how good the summaries were, only how well the different runs with SweSum corresponded to what our informants wanted to see in the summaries. That is, the figures represent how well SweSum mimics human selection with and without the use of Named Entity Recognition.

6.1 Reference Errors

The difference in readability and coherence of the two types of SweSum generated summaries was quite interesting. When scrutinizing the extracts we decided to look at a typical problem with extraction-based summarization - reference errors due to removed antecedents (i.e. dangling anaphora). This error was divided into two severity levels, anaphors that refer to the wrong antecedent and anaphors that does not have any antecedent at all to point to.

In the subset of extracts generated using Named Entity Recognition there where a total of three reference errors (pronouns etc.) and 13 cases of completely lost context over the ten extract summaries (see Table 2). In the summaries generated not using Named Entity Recognition there were six reference errors and only two cases of completely lost context over the ten summaries.

	With NER	**Without NER**
Reference errors	3 errors	6 errors
Complete loss of context	13 cases	2 cases

Table 2. Referential errors in Group 1 extracts.

The extracts generated using Named Entity Recognition clearly showed a lot more coherency problems and loss of context.

To verify the above observations and to see how much the weighting of named entities affected the summarization result we collected a second set of texts (Group 2) and generated new summaries. The second set consisted of 10 news texts randomly chosen from KTH News Corpus (Hassel 2001). These were summarized with a high, low and no weight on named entities in SweSum. As shown in Table 3 the observations for the Group 1 summaries were very much verified in Group 2. In this new set of extract summaries those generated using Named Entity Recognition showcased a total of 10 respectively 12 reference errors while the set of summaries generated not using Named Entity Recognition only contained 4 errors over the ten summaries.

NE weighting	**High weight**	**Low weight**	**No weight**
Reference errors	3 errors	3 errors	2 errors
Complete loss of context	7 cases	9 cases	2 cases

Table 3. Referential errors in Group 2 extracts.

Surprisingly enough the gold standard showed no reference error at all.

6.2 Loss of Background Information

Our conclusion is that weighting of named entities tend to prioritize singular sentences high in information centered on the categories used. The result is that it tends to prioritize elaborative sentences over introductory and thus sometimes is responsible for serious losses of sentences giving background information. Our guess is that elaborative sentences have more named entities per sentence than introductory due to the fact that introductory sentences focus on something newly introduced in the text. However we have no statistics to substantiate this claim. This often lessens the coherency of the summary (see Example 1). One solution to this would of course be to extract the paragraph with the highest-ranking sentences

(Fuentes and Rodriguez 2002); another is to let sentence position highly outweigh named entities (Nobata *et al.* 2002).

> - Hennes tillstånd är livshotande, säger jourhavande åklagare **Åke Hansson**.
> Lisa **Eriksson** var knapphändig i sina uppgifter på tisdagen.
> Sjukvården i *Sundsvall* räckte inte till för att rädda flickan.
> Enligt läkare i *Uppsala* var hennes tillstånd i går fortfarande livshotande.
> 2001 anmäldes nära 7 000 fall av barnmisshandel i *Sverige*. På **Astrid** Lindgrens barnsjukhus i *Solna* upptäcks i dag ungefär ett spädbarn i månaden som är offer för den form av barnmisshandel som kallas Shaken baby-syndrome.
> **Petter Ovander**
> **Example 1.** Summarized with weighting of named entities

One way of bouting the problem of loss of background information is of course to raise the size of the extraction unit. If we raise the extraction unit to encompass for example paragraphs instead of sentences the system would identify and extract only the most important paragraph(s) as in (Fuentes and Rodriguez 2002). This would lessen the risk of loosing background information at least on paragraph level as well as almost completely eliminate the risk of dangling anaphora for extracted pronouns. On longer texts loss of background information and coherency problem can still of course arise on chapter or text level.

Another way to try to benefit from the use of Named Entity Recognition in Automatic Text Summarization without risking the loss of background information is of course to use a very low weight for NE relative to other weights used (for example keyword frequency and sentence position) and hope that it fine-tunes the summary rather than letting it have a large negative impact on it. This is supported by experiments by Nobata *et al.* (2002) where they trained an automatic summarization system on English {extract,text} tuples and noted that the weight given by the training system to the Named Entity Recognition module was significantly lower than for the other modules.

6.3 Condensed Redundancy

When no weighting of named entities is carried out clusters of interrelated sentences tend to get extracted because of the large amount of common words. This gives high cohesion throughout the summary but sometimes leads problems with condensed redundancy. For example:

> 6 veckors baby svårt misshandlad
> Pappan misstänkt för misshandeln
> En sex veckor gammal bebis kom sent i lördags kväll svårt misshandlad in på akuten i Sundsvall. Flickan har mycket svåra skall- och lungskador. - Hennes tillstånd är livshotande, säger jourhavande åklagare Åke Hansson. Barnets pappa har anhållits som misstänkt för misshandeln på den sex veckor gamla flickan.
> Sex veckor gammal
> Flickan - som enligt uppgift till Aftonbladet är sex veckor gammal - kom in till akuten Sundsvalls sjukhus vid 22-tiden i lördags kväll. Hennes skador var livshotande.
> Petter Ovander

Example 2. Summarized without weighting of named entities

We can clearly see how redundancy in the original text "sex veckor gammal" ("six weeks old") is not only preserved but rather emphasized in the summary. This is because the term frequency (*tf*), the frequency of the keywords, heavy influences the selection.

6.4 Over-explicitness

When summarizing with weighting of named entities the resulting summaries sometimes seem very repetitive (see Example 3) but are in fact generally less redundant than the ones created without weighting of named entities.

> Pojkarna skrek att de ville ha pengar och beordrade **Pierre** att gå till kassan.
> **Pierre** minns inte i detalj vad som sedan hände, mer än att det första yxhugget träffade I ryggen.
> Liggande på marken fick **Pierre** ta emot tre yxhugg i huvudet.
> **Pierre** lyckades slita yxan ur händerna på 28-åringen.
> **Pierre** hade svårt att läsa och fick börja om från början igen.
> I dag har **Pierre** lämnat händelserna 1990 bakom sig.
> Psykiskt har **Pierre** klarat sig bra.

Example 3. Summarized with weighting of named entities

In this case the male name Pierre is repeated over and over again. With the proper noun repeated in every sentence the text appears overly explicit and staccato like. There is no natural flow and the text feels strained and affected. A solution to this would be to generate pronouns in short sequences and keeping only for example every third occurrence of a name in an unbroken name-dropping sequence.

7 Conclusions

Named entities, as well as high frequent keywords, clearly carry clues to the topic of a text. Named entities tend to identify informative extraction segments without emphasizing redundancy by preferring similar segments. A major problem we identified in our experiments is that the Named Entity module tends to prioritize elaborative sentences over introductory and thus sometimes is responsible for serious losses of sentences giving background information. Because of this one of the main difficulties using named entities in the weighting scheme would be, as with any lexical or discourse parameter, how to weight it relatively the other parameters. When centering the summary on a specific Named Entity there also arises the need for pronoun generation to avoid staccato like summaries due to over-explicitness.

When producing informative summaries for immediate consumption, for example in a news surveillance or business intelligence system, the background may often be more or less well known. In this case the most important parts of the text is what is new and which participants play a role in the scenario. Here Named Entity Recognition can be helpful in highlighting the different participants and their respective role in the text. Other suggested and applied methods of solving the coherence problem are, as we have seen, to raise the extraction unit to the level of paragraphs or to use a very low, almost insignificant, weight on named entities.

8 Demonstrators

The two different versions of SweSum as well as the small corpus of Swedish news texts and man-made extracts are available on the web if anyone desires to reproduce or do further experiments. The corpus comes with the gold standard extracts generated by majority vote as well as three computer generated baselines. These are available on the following addresses:

SweSum (standard version):
http://swesum.nada.kth.se/index-eng.html
SweSum (NE version):
http://www.nada.kth.se/˜xmartin/swesum_lab/index-eng.html
KTH extract corpus:
http://www.nada.kth.se/iplab/hlt/kthxc/showsumstats.php

SweNam is also available online for testing purposes at:
http://www.nada.kth.se/˜xmartin/swene/index-eng.html

References

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical Report TRITA-NA-P0015, IPLab-174, KTH NADA, Sweden.

Hercules Dalianis and Erik Åström. 2001. SweNam - A Swedish Named Entity recognizer. Its construction, training and evaluation. Technical Report TRITA-NA-P0113, IPLab-189, KTH NADA, Sweden.

Harold Parkins Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Maria Fuentes and Horacio Rodriguez. 2002. Using cohesive properties of text for Automatic Summarization. In *JOTRI2002 - Workshop on Processing and Information Retrieval*.

Martin Hassel. 2001. Internet as Corpus - Automatic Construction of a Swedish News Corpus. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

Eduard Hovy and Chin-Yew Lin. 1997. Automated Text Summarization in SUMMARIST. In *Proceedings of the ACL97/EACL97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, July 1997.

Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development*, 2(2):159–165.

Inderjeet Mani. 2001. Summarization Evaluation: An Overview. In *Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization*.

Inderjeet Mani and Mark T. Maybury, editors. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Chikashi Nobata, Satoshi Sekine, H. Isahara, and R. Grishman. 2002. Summarization System Integrated with Named Entity Tagging and IE pattern Discovery. In *Proceedings of Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Canary Islands, Spain.

Gerard Salton. 1988. *Automatic Text Processing*. Addison-Wesley Publishing Company.

Paper 4

Generation of Reference Summaries

Martin Hassel and Hercules Dalianis. 2005. *Generation of Reference Summaries.* In Proceedings of 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznan, Poland, April 21–23 2005.

Generation of Reference Summaries

Martin Hassel

IPLab, KTH KOD
xmartin@nada.kth.se

Hercules Dalianis

DSV-KTH / Stockholm University
hercules@kth.se

Abstract

We have constructed an integrated web-based system for collection of extract-based corpora and for evaluation of summaries and summarization systems. During evaluation and examination of the collected and generated data we found that in a situation of low agreement among the informants the corpus gives unduly favors to summarization systems that use sentence position as a central weighting feature. The problem is discussed and a possible solution is outlined.

1. Background

When developing text summarizers and other information extraction tools it is extremely difficult to assess the performance of these tools. One reason for this is that evaluation is time-consuming and needs large manual efforts. When changing the architecture of the summarizer one needs to carry out the evaluation process again.

Therefore it would be fruitful to have a tool that directly can assess the result from a text summarizer repeatedly and automatically. We have for this reason constructed the KTH extract tool to create an extract corpus that can be used to evaluate text summarizers.

To create the extract corpus we need a large group of human informants. When the extract corpus is in place it can be used repeatedly with little effort. One other advantage is that one can create an extract corpus in any language and evaluate any language-dependant text summarizer, as long as one is sure about the quality of the corpus. In order to use the extract corpus for evaluation of a summarizer one needs careful preparation of the corpus, also it is important to discuss in what sense the extract corpus can correspond to the output of the summarizer.

The specific target for our evaluation is the SweSum text summarizer for Swedish news text and the DanSum[1] text summarizer for Danish news text.

SweSum is a text summarizer mainly developed to summarize Swedish news text (Dalianis 2000). SweSum works on sentence level – i.e. extracting sentences, judging the relevance of each sentence and then creating a shorter text (non-redundant extract) containing the highest-ranking sentences from the original text.

SweSum has been ported to English, Spanish, French, Danish, Norwegian, German and Farsi so far. SweSum is freely available online at http://swesum.nada.kth.se, and we have today around 2 200 visitors per month using it, mostly from American and Spanish universities.

1.1 Previous Research

Evaluating summaries and automatic text summarization systems is not a straightforward process. What exactly makes a summary beneficial is an elusive property. Generally speaking there are two properties of the summary that must be measured when evaluating summaries and summarization systems: the Compression Ratio (how much shorter the summary is than the original);

CR = length of Summary / length of Full Text

and the Retension Ratio (how much information is retained);

RR = information in Summary / information in Full Text

Retention Ratio is sometimes also referred to as Omission Ratio, (Hovy 1999). An evaluation of a summarization system must at least in some way tackle both properties.

1.2 Evaluation methods

A first broad division in methods for evaluation automatic text summarization systems, as well as many other systems, is into intrinsic and extrinsic evaluation methods (Spark-Jones and Galliers 1995).

Extrinsic evaluation measures the efficiency and acceptability of the generated summaries in some task, for example relevance assessment or reading comprehension.

Intrinsic evaluation on the other hand measures the system in of itself. This is often done by comparison to some gold standard, which can be made by a reference summarization system or, more often than not, is man-made using informants. Intrinsic evaluation has mainly focused on the coherence and informativeness of summaries.

Summaries generated through extraction-based methods (cut-and-paste operations on phrase, sentence or paragraph level) sometimes suffer from parts of the summary being extracted out of context, resulting in coherence problem (e.g. dangling anaphors or gaps in the rhetorical structure of the summary). One way to measure this is to let subjects rank or grade summary sentences for coherence and then compare the grades for the summary sentences with the scores for reference summaries.

For single documents traditional precision and recall figures can be used to assess performance as well as utility figures and content based methods. Precision and

[1] DanSum is SweSum ported to Danish

recall are standard measures for Information Retrieval and are often combined in a so-called F-score. The main problems with these measures for text summarization is that they are not capable of distinguishing between many possible, but possibly equally good, summaries and that summaries that differ quite a lot content wise may get very similar scores.

Sentence rank is a more fine-grained approach than precision and recall (P&R), where the reference summary is constructed by ranking the sentences in the source text by worthiness of inclusion in a summary of the text. Correlation measures can then be applied to compare the generated summary with the reference summary. As in the case of P&R this method mainly applies to extraction based summaries, even if standard methods of sentence alignment with abstracts can be applied (see Marcu 1999, Jing and McKeown 1999).

The utility method (UM) (see Radev et al. 2000) allows reference summaries to consist of extraction units (sentences, paragraphs etc.) with fuzzy membership in the reference summary. In UM the reference summary contains all the sentences of the source document(s) with confidence values for their inclusion in the summary.

This method bears many similarities to the Majority Vote method (Hassel 2003) in that it, in contrast to standard P&R and Percent Agreement, allows summaries to be evaluated at different compression rates. UM is mainly useful for evaluating extraction-based summaries; more recent evaluation experiments has led to the development of the Relative Utility metric (Radev and Tam 2003).

1.3 Evaluation Tools

We have described a number of evaluation methods, now we need tools to use these methods. These tools will support us in creating a framework for more rigorous and repeatable evaluation procedure, partly by automating the comparison of summaries.

It is advantageous to build an extract corpus containing original full texts and their corresponding extracts, i.e. summaries strictly made by extraction of, in our case, whole sentences from an original text. Each extract, whether made by a human informant or a machine, is meant to be a true summary of the original, i.e. to retain the central points of the text to as large extent as possible

A number of tools have been developed for these purposes. Summary Evaluation Environment (SEE; Lin 2001) is an evaluation environment in which assessors can evaluate the quality of a summary, called the peer text, in comparison to a reference summary, called the model text. The texts involved in the evaluation are pre-processed by being broken up into a list of segments (phrases, sentences, clauses, etc.) During the evaluation phase, the two summaries are shown in two separate panels in SEE and interfaces are provided for assessors to judge both the content and the quality of model summaries. The assessor rates each unit and the overall structure of the model summary.

MEADeval (Winkel and Radev 2002) is a Perl toolkit for evaluating MEAD- and DUC-style extracts, by comparison to a reference summary (or ”ideal” summary). MEADeval operates mainly on extract files, which describe the sentences contained in an extractive summary: which document each sentence came from and the number of each sentence within the source document – but can also perform some general content comparison. It supports a number of standard metrics, as well as some specialized

The ISI ROUGE - Automatic Summary Evaluation Package. ROUGE, short for Recall-Oriented Understudy for Gisting Evaluation, by Lin (2003). According to in-depth studies based on various statistical metrics and comparison to the results DUC-2002 (Hahn and Harman 2002), this evaluation method correlates surprisingly well with human evaluation (Lin and Hovy 2003). ROUGE is recall oriented, in contrast to the precision oriented BLEU script, and separately evaluates 1, 2, 3, and 4-grams. ROUGE has been verified for extraction-based summaries with a focus on content overlap. No correlation data for quality has been found so far.

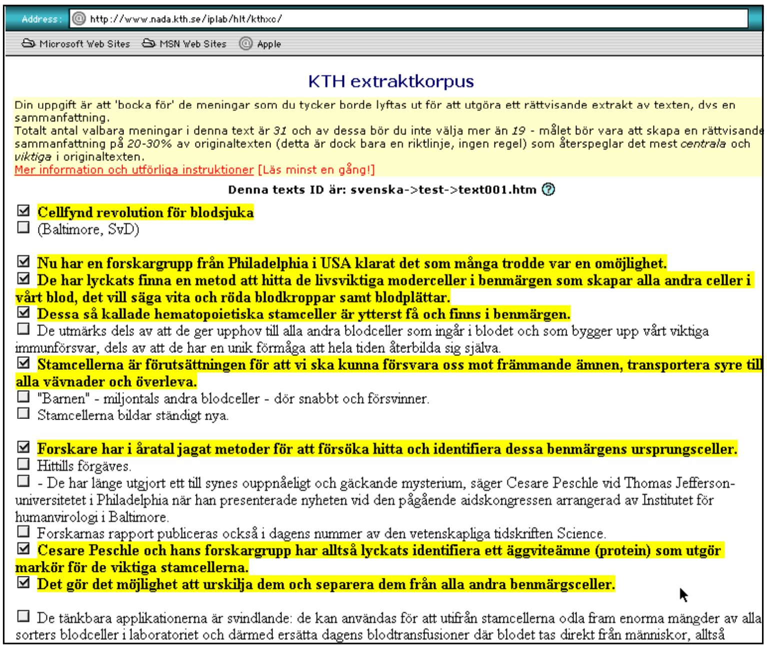
However, none of the above tools have any support to help informants to create extracts, thus aiding in corpora building as well as evaluation.

2. KTH eXtract Corpus tool

At KTH, Stockholm, the KTH eXtract Corpus tool has been constructed (Hassel 2003, Dalianis et al. 2004). The tool assists in the collection of extract-based summaries provided by human informants and semi-automatic evaluation of machine generated extracts in order to easily evaluate the SweSum summarizer (Dalianis 2000). The KTH eXtract Corpus (KTHxc) contains a number of original full texts and several man-made extracts for each text. The tool assists in the construction of an extract corpus by guiding the human informant creating a summary in such a way that only full extract units (most often sentences) are selected for inclusion in the summary, (see figure 1). The interface allows for the reviewing of sentence selection at any time, as well as reviewing of the constructed summary before submitting it to the corpus.

Once the extract corpus is compiled, the corpus can be analyzed automatically in the sense that the inclusion of sentences in the various extracts for a given source text can easily be compared. Also available is the possibility of comparison on word level, a so-called vocabulary test. This allows for a quick adjustment and evaluation cycle in the development of an automatic summarizer. One can, for instance, adjust parameters of the summarizer and directly obtain feedback of the changes in performance, instead of having a slow, manual and time-consuming evaluation.

The KTH extract tool gathers statistics on how many times a specific extract unit from a text has been included in a number of different summaries. Thus, an ideal summary, or reference summary, can be composed using only the most frequently chosen sentences.



http://www.nada.kth.se/iplab/hlt/kthxc/

KTH extraktkorpus

Din uppgift är att 'bocka för' de meningar som du tycker borde lyftas ut för att utgöra ett rättvisande extrakt av texten, dvs en sammanfattning.
Totalt antal valbara meningar i denna text är *31* och av dessa bör du inte välja mer än *19* - målet bör vara att skapa en rättvisande sammanfattning på *20-30%* av originaltexten (detta är dock bara en riktlinje, ingen regel) som återspeglar det mest *centrala* och *viktiga* i originaltexten.
Mer information och utförliga instruktioner [Läs minst en gång!]

Denna texts ID är: svenska->test->text001.htm

☑ **Cellfynd revolution för blodsjuka**
☐ (Baltimore, SvD)

☑ **Nu har en forskargrupp från Philadelphia i USA klarat det som många trodde var en omöjlighet.**
☑ **De har lyckats finna en metod att hitta de livsviktiga moderceller i benmärgen som skapar alla andra celler i vårt blod, det vill säga vita och röda blodkroppar samt blodplättar.**
☑ **Dessa så kallade hematopoietiska stamceller är ytterst få och finns i benmärgen.**
☐ De utmärks dels av att de ger upphov till alla andra blodceller som ingår i blodet och som bygger upp vårt viktiga immunförsvar, dels av att de har en unik förmåga att hela tiden återbilda sig själva.
☑ **Stamcellerna är förutsättningen för att vi ska kunna försvara oss mot främmande ämnen, transportera syre till alla vävnader och överleva.**
☐ "Barnen" - miljontals andra blodceller - dör snabbt och försvinner.
☐ Stamcellerna bildar ständigt nya.

☑ **Forskare har i åratal jagat metoder för att försöka hitta och identifiera dessa benmärgens ursprungsceller.**
☐ Hittills förgäves.
☐ - De har länge utgjort ett till synes ouppnåeligt och gäckande mysterium, säger Cesare Peschle vid Thomas Jefferson-universitetet i Philadelphia när han presenterade nyheten vid den pågående aidskongressen arrangerad av Institutet för humanvirologi i Baltimore.
☐ Forskarnas rapport publiceras också i dagens nummer av den vetenskapliga tidskriften Science.
☑ **Cesare Peschle och hans forskargrupp har alltså lyckats identifiera ett äggviteämne (protein) som utgör markör för de viktiga stamcellerna.**
☑ **Det gör det möjlighet att urskilja dem och separera dem från alla andra benmärgsceller.**

☐ De tänkbara applikationerna är svindlande: de kan användas för att utifrån stamcellerna odla fram enorma mängder av alla sorters blodceller i laboratoriet och därmed ersätta dagens blodtransfusioner där blodet tas direkt från människor, alltså

Figure 1. The KTH extract tool in action assisting an informant in creating an extract.

The reference summary can be generated at an arbitrary compression rate, i.e. the most high-ranking extract units up to a desired percentage of the original text. When several units with an equal number of votes and not all of them will fit into the reference summary units are extracted in order to prevent dangling anaphoric references.

Further statistical analysis can evaluate how close a particular extract is to a reference summary constructed by majority vote. The tool also has the ability to output reference summaries in the format SEE (see above) uses for human assessment. The KTHxc tool can easily be ported to other languages as the interface is completely separated from the code, and so far corpus collection and evaluation has been conducted for Swedish as well as Danish[2] news texts.

3. Creating the extract corpus

Three groups of texts have been collected during three iterations, two Swedish and one Danish group.

The Swedish extract corpus consists of a total of 301 Swedish text extracts submitted by 45 informants; average length of submitted extracts is currently 32.5 percent (31% and 34% respectively for group 1 and 2).

The Danish extract corpus at the present consists of 135 Danish text extracts submitted by 15 informants; average length of submitted extracts here is currently 32%.

During the extraction phase the human informants where allowed to submit extract summaries as short as 5 percent and up to 60 percent of the original text. The mean length of the submitted extracts varied between the texts, partly due to the length of the original text but also depending on the nature of the text (number of sentences, percentage of short respectively long sentences and of course also the texts rhetorical structure. However, the mean length of the submitted extracts over respectively of the three different groups was fairly consistent, ranging between 31 and 34 percent.

4. Evaluating

This experiment shows that there is not very much agreement between the informants on which sentences to select for the extract summary. The level of agreement among the informants was calculated with a simple precision function. This is done per text and then a mean value was calculated over all texts in each group.

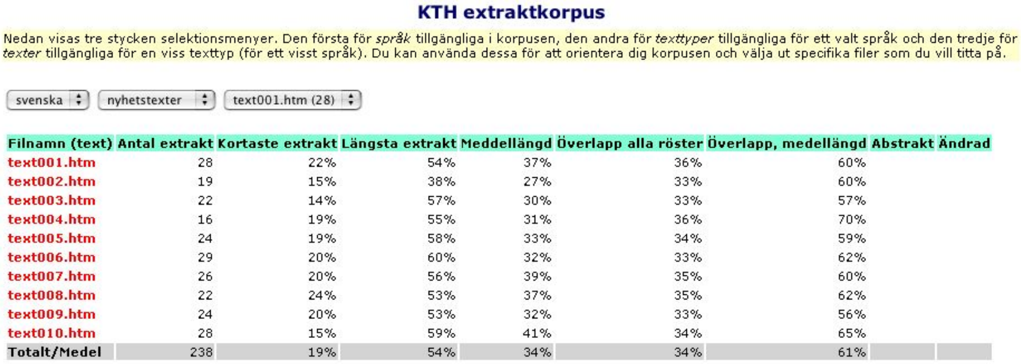
$$\frac{Vc}{Ns * Nx} * 100$$

In the function above *Vc* is the number of votes that are represented in the generated extract, *Ns* is the number of sentences represented in the same extract and *Nx* is the number of manmade extracts made for the original text the votes and sentences account for. This means that when all informants choose not only the same number of sentences but also exactly the same set of sentences the function will result in a precision, or agreement, of 100%.

We were prepared for a low agreement among the human extractors as to which sentences are good summary sentences as previous studies have shown this (for an overview see Mani 2001). In a previous study by Hassel (2003) using 11 informants and 96 extracts we found that when taking all selected extraction units into account for each text that there was only a mean agreement of 39.6% over ten texts.

This is however not so bad as it can seem at first glance. When generating a “gold standard” summary by presenting the most selected sentences up to a length of the mean length of all submitted extracts for a given text the precision, or the agreement level, rose to 68.9%. Very few of the sentences chosen for the gold standard where selected by as few as one third or less of the informants. Of course, even fewer sentences where selected by all

[2] The University of Bergen has initiated a similar effort for Norwegian and has developed some similar tools (Dalianis et al. 2004).



KTH extraktkorpus

Nedan visas tre stycken selektionsmenyer. Den första för *språk* tillgängliga i korpusen, den andra för *texttyper* tillgängliga för ett valt språk och den tredje för *texter* tillgängliga för en viss texttyp (för ett visst språk). Du kan använda dessa för att orientera dig korpusen och välja ut specifika filer som du vill titta på.

svenska | nyhetstexter | text001.htm (28)

Filnamn (text)	Antal extrakt	Kortaste extrakt	Längsta extrakt	Meddellängd	Överlapp alla röster	Överlapp, medellängd	Abstrakt	Ändrad
text001.htm	28	22%	54%	37%	36%	60%		
text002.htm	19	15%	38%	27%	33%	60%		
text003.htm	22	14%	57%	30%	33%	57%		
text004.htm	16	19%	55%	31%	36%	70%		
text005.htm	24	19%	58%	33%	34%	59%		
text006.htm	29	20%	60%	32%	33%	62%		
text007.htm	26	20%	56%	39%	35%	60%		
text008.htm	22	24%	53%	37%	35%	62%		
text009.htm	24	20%	53%	32%	33%	56%		
text010.htm	28	15%	59%	41%	34%	65%		
Totalt/Medel	238	19%	54%	34%	34%	61%		

Figure 2.Overview of the web-based statistics of the extract corpus for Swedish (the text above is in Swedish)

informants. In fact, not even all informants could agree upon extracting the title or not when one was present.

Later we obtained more informants in form of students from our courses and we found that mean agreement decreased to 34% for all selected sentences and the mean agreement down to 61% for texts of summary length of the mean length of all man-made extracts, see figure 2.

These results are somewhat agreeing with work in manual indexing of texts (Bäckström 2000, van Dijk 1995). Bäckström found only 30 percent agreement, or index consistency, in selecting index terms in Swedish between two inexperienced human indexers and van Dijk (in French) found 60-80 percent agreement between two experienced indexers.

In order to verify this relationship between non-experienced and experienced informants we collected a second set of extracts using language consultant students as informants. This group has shown an agreement level of 73 percent when the most selected sentences up to a summary length of the mean length of all submitted extracts for each text. This is the highest agreement level of the three groups. This is probably the case since language consultants are well-trained readers and writers.

The extract summaries generated with SweSum[3] where then semi-automatically[4] compared on sentence and word level with the gold standard extracts generated by majority vote. We found that the summaries generated by SweSum and the gold standard summaries had between 47 and 62 percent of the sentences in common.

Of course this does not say much about how useful or coherent the system generated summaries were, only how well the different summaries created by SweSum corresponded to what our informants wanted to see in the summaries. That is, the figures represent how well SweSum mimics human selection.

However, what we find striking is the fact that SweSum apparently performs worse in regards to the reference summary when the agreement level rises. The reason for this seems to be that when the agreement level is high, which means that most votes are concentrated on a few sentences; it is less probable that the summarizer by "chance" hits the selected sentences. If, on the other hand, the agreement level is low and the votes are more evenly spread over the sentences, it might be the case that SweSum has a higher chance of hitting the same sentences as in the reference summary, since both system solve ties[5] by prioritizing the sentence occurring earliest in the text.

What we have here is a case of an evaluation method that evaluates partly along the same premises as the system it is evaluating. A system that does not put a high focus on sentence position might not get scored as favorably. This might, for example, be one reason that SweSum scores better than the Spanish lexical chain summarizer in the system-to-system comparison against Spanish model summaries made by Alonso i Alemany and Fuentes Fort (2003).

5. Tie Breaking

A more intelligent tie breaking scheme is clearly in need, preferably one that relies more on submitted data than on a general method that might be exploited by summarization system to be evaluated. One such is what we could call mutual exclusion; another could be called mutual inclusion.

Mutual exclusion as a tie breaking method would occur when in the statistics two or more sentences, or extract units, that have received the same number of votes show no informant overlap in the statistics. That is, when no, or very few, informants who have chosen one sentence have also chosen another we can assume that there is a reason for this, for example information redundancy.

Mutual inclusion would, on the other hand, occur when all, or almost all, informants have chosen the same set of

[3] The extracts where generated with SweSum by setting the desired compression rate to equal the mean length of all submitted man-made extracts for each text.

[4] The SweSum generated extracts where pasted into the evaluation view of the corpus interface.

[5] A tie is here defined as when two or more extract units receive equal score or, in the case of the reference summary, selection frequency.

sentences. This means that if a local high agreement occurs within the text this bond should be preserved.

6. Conclusions

In automatic text summarization, as well as in for example machine translation, there may be several equally good summaries for one specific source text effectively making evaluation against one rigid reference summary unsatisfactory. Also, evaluation methods that allow for evaluation at different compression rates should be favored, as experiments have shown that different compression rates are optimal for different text types or genres, or even different texts within a text type or genre. The semi-automatic evaluation methods presented in this paper attempts to tackle these properties. The described system mainly deals with content similarity between summaries. Summary quality, i.e. cohesion and coherence, must still be evaluated manually.

However, as we have shown, counting votes is not enough when constructing extraction-based corpora from many extracts. The distribution of the votes should also be taken into account in order to extract text binding clues hidden in the distribution of the votes.

Also, when generating the reference summaries from the selection statistics one must be aware of how the system solves tie breaking in case of equal number of votes and how this may favor the summarization system being evaluated.

Acknowledgements
We would express gratitude to all the informants who have been willing to submit time and effort in compiling the selection statistics. We would also like to thank the participants of the ScandSum network and especially professor Koenraad de Smedt and Anja Liseth of the University of Bergen, Norway, for fruitful and invigorating discussions on summarization and evaluation.

References

Alonso i Alemany, L. and M. Fuentes Fort (2003). Integrating cohesion and coherence for automatic summarization. In Proceedings of EACL 2003, Budapest, Hungary.

Bäckström, K. 2000. Marknadsundersökning och utvärdering av indexeringsprogram – en delstudie inom projektet Automatisk Indexering Magisteruppsats vid Institutionen för lingvistik Uppsala Universitet. (Master Thesis in Swedish)

Dalianis, H. 2000. Swesum - a text summarizer for Swedish. Technical report TRITA-NA-P0015, IPLab-174 NADA, KTH, Sweden.

Dalianis, H., M. Hassel, K. de Smedt, A. Liseth, T. C. Lech, and J. Wedekind. 2004. Porting and evaluation of automatic summarization. In H. Holmboe (editor), Nordisk Sprogteknologi 2003: Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004. Museum Tusculanums Forlag.

van Dijk, B. 1995. Parlement Européen:Evaluation des operations pilotes d'indexation automatique. (Convention spécifique no 52556) Rapport d'evaluation finale. (in French)

Donaway, R. L., K. W. Drummey, and L. A. Mather 2000.. A Comparison of Rankings Produced by Summarization Evaluation Measures. In U. Hahn, C.-Y. Lin, I. Mani, and D. R. Radev (editors), Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and NAACL 2000.

Hahn, U- and D. Harman (editors) 2002. Proceedings of the 2nd Document Understanding Conference. Philadelphia, PA.

Hassel, M. 2003. Exploitation of Named Entities in Automatic Text Summarization for Swedish. In In the Proceedings of NODALIDA'03 - 14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland.

Jing, H. and K. R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In M. Hearst, G. F., and R. Tong (editors), Proceedings ofthe 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 129–136, University of California, Beekely.

Lin, C.-Y. 2001. Summary Evaluation Environment. http://www.isi.edu/˜cyl/SEE.

Lin, C.-Y. 2003. ROUGE: Recall-oriented understudy for gisting evaluation. http://www.isi.edu/˜ cyl/ROUGE/.

Lin, C.-Y. and E. Hovy. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada.

Mani, I. 2001. Summarization Evaluation: An Overview. In Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization.

Marcu, D. .1999. The Automatic Construction of Large-Scale Corpora for Summarization Research. In M. Hearst, G. F., and R. Tong (editors), Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 137–144, University of California, Berkely.

Radev, D. R., H. Jing, and M. Budzikowska. 2000. Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies. In U. Hahn, C.-Y. Lin, I. Mani, and D. R. Radev (editors), Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and NAACL 2000, Seattle, WA.

Spark-Jones, K. and J. R. Galliers. 1995. Evaluating Natural Language Processing Systems: An Analysis and Review. Number 1083 in Lecture Notes in Artificial Intelligence. Springer.

Winkel, A. and D. Radev. 2002. MEADeval: An evaluation framework for extractive summarization. http://perun.si.umich.edu/clair/meadeval/.

Appendix

Swedish Group 1					Comparison with best extract			Comparison with SweSum summary		
	Extracts	Average	Overlap	Overlap	Overlap at			Overlap at		
Extracts	No of	extract	of	at mean	sentence	word	word	sentence	word	word
Filename	extracts	length	all votes	length	level	level	frequency	level	level	frequency
text001.htm	28	37%	36%	60%	80%	83%	71%	80%	81%	75%
text002.htm	19	27%	33%	60%	88%	81%	74%	46%	53%	42%
text003.htm	22	30%	33%	57%	86%	94%	90%	83%	94%	90%
text004.htm	16	31%	36%	70%	63%	75%	69%	78%	84%	79%
text005.htm	24	33%	34%	59%	74%	84%	77%	59%	67%	63%
text006.htm	29	32%	33%	62%	77%	78%	74%	44%	64%	53%
text007.htm	26	39%	35%	60%	86%	94%	90%	75%	70%	65%
text008.htm	22	37%	35%	62%	67%	77%	72%	36%	63%	56%
text009.htm	24	32%	33%	56%	61%	68%	65%	59%	79%	70%
text010.htm	28	41%	34%	65%	80%	92%	92%	57%	75%	66%
Total/Average	238	34%	34%	61%	76%	82%	77%	62%	73%	66%

Table 1: Agreement among Swedish colleagues and students, comparison of "best" submitted extract with majority vote extract and comparison of SweSum generated extract with Majority Vote extract.

Swedish Group 2					Comparison with best extract			Comparison with SweSum summary		
	Extracts	Average	Overlap	Overlap	Overlap at			Overlap at		
Extracts	No of	extract	of	at mean	sentence	word	word	sentence	Word	word
Filename	extracts	length	all votes	length	level	level	frequency	level	Level	frequency
text001.htm	12	33%	48%	79%	89%	93%	89%	50%	62%	52%
text002.htm	11	22%	39%	69%	74%	80%	75%	34%	43%	33%
text003.htm	11	34%	38%	69%	94%	98%	97%	63%	70%	59%
text004.htm	15	32%	43%	80%	100%	100%	100%	29%	31%	23%
text005.htm	14	32%	42%	67%	78%	86%	78%	57%	70%	61%
Total/Average	63	31%	42%	73%	87%	92%	88%	47%	55%	46%

Table 2: Agreement among Swedish language consultant students, comparison of "best" submitted extract with majority vote extract and comparison of SweSum generated extract with Majority Vote extract.

Danish Group					Comparison with best extract			Comparison with SweSum summary		
		Average	Overlap	Overlap	Overlap at			Overlap at		
Extracts	No of	extract	of	at mean	sentence	word	word	sentence	word	word
Filename	extracts	length	all votes	length	level	level	frequency	level	level	frequency
text001.htm	15	34%	44%	76%	83%	90%	87%	57%	77%	67%
text002.htm	14	35%	40%	66%	89%	95%	92%	67%	56%	47%
text003.htm	15	32%	34%	67%	91%	97%	95%	18%	56%	43%
text004.htm	12	23%	32%	55%	64%	70%	59%	44%	55%	44%
text005.htm	13	30%	39%	60%	86%	92%	87%	57%	54%	43%
text006.htm	12	28%	37%	67%	89%	88%	81%	60%	68%	57%
text007.htm	11	33%	42%	63%	93%	96%	94%	67%	69%	63%
text008.htm	14	37%	31%	79%	67%	76%	69%	40%	46%	31%
text009.htm	15	38%	37%	73%	100%	100%	100%	67%	81%	71%
text010.htm	14	31%	38%	65%	100%	100%	100%	40%	73%	65%
Total/Average	135	32%	37%	67%	86%	90%	86%	52%	64%	53%

Table 3: Agreement among colleagues at CST, Denmark, comparison of "best" submitted extract with majority vote extract and comparison of SweSum generated extract with Majority Vote extract.

Paper 5

Navigating Through Summary Space – Selecting Summaries, Not Sentences

Martin Hassel and Jonas Sjöbergh. Submitted 2007. *Navigating Through Summary Space – Selecting Summaries, Not Sentences.*

Navigating Through Summary Space - Selecting Summaries, Not Sentences

Martin Hassel and Jonas Sjöbergh
KTH CSC
Royal Institute of Technology
100 44 Stockholm, Sweden
{xmartin,jsh}@nada.kth.se

Abstract

We present a novel method for extraction based summarization using statistical lexical semantics. It attempts to give an overview by selecting the summary most similar to the source text from a set of possible candidates. It evaluates whole summaries at once, making no judgments on for instance individual sentences. A simple greedy search strategy can be used to search through a space of possible summaries. Starting the search with the leading sentences of the source text is a powerful heuristic, but we also evaluate other search strategies. The aim has been to construct a summarizer that can be quickly assembled, with the use of only a very few basic language tools. The proposed method is largely language independent and can be used even for languages that lack large amounts of structured or annotated data, or advanced tools for linguistic processing. When evaluated on English abstracts from the Document Understanding Conferences it performs well, though better language specific systems are available. It performs better than several of the systems evaluated there, but worse than the best systems. We have also evaluated our method on a corpus of human made extracts in Swedish. It performed poorly compared to a traditional extraction-based summarizer. However, since these man-made extracts were not produced to reflect the whole contents of the texts, but rather to cover only the main topic, this was expected.

1 Introduction

Summaries are an important tool when familiarizing oneself with a new subject area. They are also essential when deciding whether reading a document in whole is necessary or not. In other words, summaries save time in daily life and work. To write a summary of a text is a non-trivial process. The contents of the text itself should be analyzed and the most central information should be extracted. The intended readers should also be considered, taking into account what knowledge they already have, possible special interests and so on. Today numerous documents, papers, reports and articles are available in digital form, most of which lack summaries. The information is often too abundant for it to be possible to sift through it manually and choose what information to acquire. The information must instead be automatically filtered and extracted to avoid drowning in it.

Automatic text summarization is a technique where a computer summarizes a text. A text is given to the computer and the computer returns a shorter, less redundant extract of the original text. So far automatic text summarization has not yet reached the quality possible with manual summarization, where a human interprets the text and writes a completely new shorter text with new lexical and syntactic choices, and may never do. However, automatic text summarization is untiring, consistent and always available.

1.1 Language Independent Automatic Text Summarization

Today most research in automatic text summarization is focused on knowledge rich, and in practice language specific, approaches using tools and annotated resources simply not available for many languages. Justifiably so, these knowledge rich systems do in general perform better than earlier knowledge poor approaches. It is however easy to see that there is a clear need for automatic summarization also for the languages less in focus in this research area than the major European, Asian or Mid Eastern languages.

The experiments reported herein concern an attempt to develop such a method for largely language independent automatic text summarization. The aim has been to construct a summarizer that can be quickly assembled, with the use of only a few basic language tools, for languages that lack large amounts of structured or annotated data or advanced tools for linguistic processing. We try to accomplish this by trying to capture the essence of a document being summarized. For this we use computational semantics by first building semantic, or conceptual, representations for each word based on a large free-text corpus. Simply put, a word space. These conceptual representations in turn are then used to build a document space where a set of summaries can be evaluated against the original text.

2 Word Spaces

Word space models, most notably Latent Semantic Analysis/Indexing (Deerwester *et al.* 1990, Landauer *et al.* 1998), enjoy considerable attention in current research on computational semantics. Since its introduction in 1990 Latent Semantic Analysis (LSA) has more or less spawned an entire research field with a wide range of word space models as a result, and numerous publications reporting exceptional results in many different tasks, such as information retrieval, various semantic knowledge tests such as the TOEFL test (Educational Testing Service 2006), text categorization and word sense disambiguation.

The general idea behind word space models is to use statistics on word distributions in order to generate a high-dimensional vector space. In this vector space the words are represented by context vectors whose relative directions are assumed to indicate semantic similarity. The basis of this assumption is the *distributional hypothesis* (Harris 1968), according to which words that occur in similar contexts also tend to have similar properties (meanings/functions). From this follows that if we repeatedly observe two words in the same, or very similar, contexts, then it is not too far fetched to assume that they also mean similar things (Sahlgren 2006).

2.1 Random Indexing

In the major part of the experiments herein we have employed Random Indexing (Sahlgren 2005), which presents an efficient, scalable and inherently incremental alternative to standard word space methods. As an alternative to LSA-like models that first construct a huge cooccurrence matrix and then use a separate dimension reduction phase, Random Indexing (RI) instead accumulates context vectors on-the-fly based on the occurrence of words (tokens) in contexts, without a need for a separate dimension reduction phase.

The construction of context vectors using RI can be viewed as a two-step operation. First, each token in the data is assigned a unique and (usually) randomly generated label. These labels can be viewed as sparse, high-dimensional, and ternary vectors.[1] Their dimensionality (d) is usually chosen to be in the range of a couple of hundred up to several thousands, depending on the size and redundancy of the data. They consist of a very small number, usually about 1-2%, of randomly distributed +1s and -1s, with the rest of the elements of the vectors set to 0.

Next, the actual context vectors are produced by scanning through the text and each time a token w occurs in a context (e.g. in a document or paragraph, or as a

[1] The extremely sparse random labels are handled internally as short lists of positions for non-zero elements, and are generated on-the-fly whenever a never before seen token is encountered in the context during indexing.

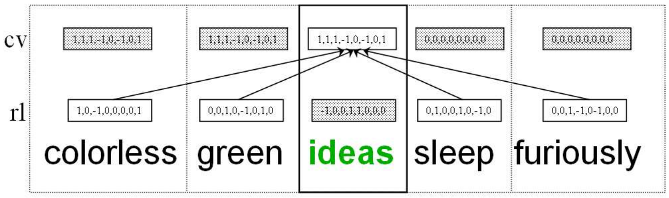


Figure 1: A Random Indexing context window focused on the token “ideas”, taking note of the cooccurring tokens. The row marked as `cv` represents the continuously updated *context vectors*, and the row marked as `rl` the static *random labels* (acting as addable meta words). Grayed-out fields are not involved in the current token update.

word within a sliding context window), that context's d-dimensional random label is added to the context vector for the token w. We use a sliding context window, i.e. all tokens that appear within the context window contribute to some degree with their random labels to the context vector for w. Words are in this way effectively represented by d-dimensional context vectors that are the sum of the random labels of the cooccurring words, see Figure 1. When using a sliding context window it is also common to use some kind of distance weighting in order to give more weight to tokens closer in context.

This technique can readily be used with any type of linguistic context and can be used to index using a more traditional bag-of-words approach as well as using a sliding context window (i.e. cooccurrence between tokens) capturing sequential relations between tokens. These tokens can be the word simply represented by its lexical string or its lemma, or more elaborate approaches utilizing tagging, chunking, parsing or other linguistic units can be employed.

One of the strengths of Random Indexing is that we can in a very elegant way fold the document currently being processed into the Random Index, thus immediately taking advantage of, possibly genre or text type specific, distributional patterns within the current document. Apart from the advantage of eliminating the risk of lack of data due to unknown words, we also have a system that learns over time. The problem of sparse data cannot be completely avoided, since a never before seen word will only have as many contextual updates as the number of times it occurs in the current document. This is however far better than no updates at all.

As with all LSA-like models Random Indexing needs, for good performance, large amounts of text (millions of words) when generating the conceptual representations. Since Random Indexing is resource lean and only requires access to raw (unannotated) text, this is generally not a problem.

There are a few implementations of Random Indexing available. We used a freely available tool-kit called JavaSDM (Hassel 2006). It should be noted that the proposed method, at least in theory, could employ any word space model, such as LSA or Hyperspace Analogue to Language (Lund *et al.* 1995), albeit waiving some of the benefits of using RI in this context.

3 Experimental Setup

The main part of these experiments have been carried out for English. Mainly because there is a large amount of reference summaries and evaluation schemes developed for this language, as well as several other summarization systems to use as reference points. For English we build our conceptual representations for each word based on a large corpus, the British National Corpus (Burnard 1995), as well as the documents themselves as they are being summarized. The data being used for building these representations thus is comprised of 100 million words from BNC and roughly 2 million words contained in 291 document sets provided for DUC 2001-2004 (DUC 2007). After stop word filtering and stemming this results in almost 290,000 unique stems taken from 4415 documents.

A minor experiment has also been carried out for Swedish in order to test the thesis of language independence. One must however keep in mind that the obvious lack of suitable and fairly large evaluation corpora render these results less reliable than their English counterparts. These results are nevertheless reported below.

3.1 Preliminary Experiment: Selecting Sentences

The first approach in our series of experiments was to build a context vector for each extraction unit, in this case each sentence, in the text being summarized. This was done by adding the context vectors for each token (word) in each individual sentence. This was also done for the complete text. All sentence vectors were then compared for similarity using the cosine angle between each sentence vector and the document vector, and the closest match was chosen. The words in the chosen sentence were then temporarily removed from the remaining sentences and their respective content vectors recalculated, and the closest match again chosen for inclusion in the summary. This procedure was repeated until the summary reached the desired length.

Different weighting and normalization schemes were tested, for example sentence length normalization and only counting each occurrence of a word in a sentence once. None of these strategies did however beat the chosen baseline summary - the first *N* sentences up to the desired summary length.[2]

This approach does, in practice, not differ particularly from most traditional extractive summarization approaches in the respect that it ranks individual extract segments for inclusion in the concatenated summary. Another criteria for selecting extraction units, using our measure of semantic similarity, was clearly in need.

3.2 Selecting Summaries: The Basic Method for English

After the preliminary experiment, we instead focused on finding summaries of a given length that are as similar to the original texts as possible. This method would aim at producing overview summaries. One way to accomplish this would be to generate all possible extracts and see which one is most similar to the original text. Besides being computationally cumbersome, the difficulty here lies in judging how similar two texts are. Most methods that compare two documents use measures like word or n-gram overlap. Since all candidate summaries here are extracts from the original text, all words in all summaries overlap with the original text. This is thus not a good way to differentiate between different candidates.

3.2.1 Evaluating Candidate Summaries

Our method makes use of Random Indexing to differentiate between different summaries. As described above, Random Indexing gives each word a context vector that in some sense represents the semantic content of that word, as defined by its use. We make use of these vectors when calculating a measure of similarity between two texts. Each text is assigned its own vector for semantic content, which is simply the (weighted) sum of all the context vectors of the words in the text. This can be seen as projecting the texts into a high-dimensional vector space where we can relate the texts to each other. Similarity between two texts is then measured as the similarity between the directions of the semantic vectors of the texts, in our case between the vector for the full text and the vectors for each of the candidate summaries. Similar approaches have also been applied to for instance text categorization (Sahlgren and Cöster 2004).

When constructing the semantic vector for a text, the context vector for each word is weighted with the term frequency and the inverse document frequency, by making the length of the vector be $tf \cdot \log(idf)$. If desired, other weighting criteria can easily be added, for instance for slanted or query based summaries where some words are deemed more important, or by giving words occurring early in the document, in document or paragraph headings etc. higher weight.

[2] This baseline summary is often referred to as *lead*.

Words in a text that have never been encountered during the calculation of a word space representation generally degrade the performance, since no information regarding their distributional properties is available at run-time. Since RI allows for continuous updates this is here trivially solved by simply adding the new text to the index immediately before summarizing it. This means in effect that all words in the relevant texts will have been encountered at least once.

Also, since our method does not give any consideration to the position in the text a sentence is taken from (though that is possible to do if one so wishes), it is relatively straightforward to use for multidocument summarization as well. In fact, some of the reference summaries in the English evaluation corpus have been built from multiple news texts covering the same event. In this case we have used the same set of source documents concatenated into one single document sent to the summarizer.

In the following section we present an extraction based technique to generate a set of summary candidates. However, the method for differentiating between the summary candidates does not require that the candidates consist solely of segments from the source text. Since the comparison of the semantic vectors does not measure lexical or syntactic similarity, but attempts to optimize semantic similarity between the summary and the text being summarized, the summary candidates could in practice be generated by any means, even being man-made.

3.2.2 Finding a Better Summary

To find a good summary we start with one summary and then try to see if there is another summary that is "close" in some sense, that is also a better summary. Better in this context means more similar to the original text. The reason we do not exhaustively pursue the best summary of all possible summaries is that there are exponentially many possible summaries. Comparing all of them to the original text would thus not be feasible even for documents with fairly few extraction units (in our case sentences).

It has been shown that the leading sentences of an article, especially within the news domain, are important and constitute a good summary (Zechner 1996). Therefore, the "lead" summary, i.e. the first sentences from the document being summarized up to a specified length, was used in our experiments both as a baseline and as the starting point for our search for a better summary. When used for multidocument summarization we simply take the concatenated set of documents covering the same topic as source text and the leading sentences of the top-most document as the starting point.

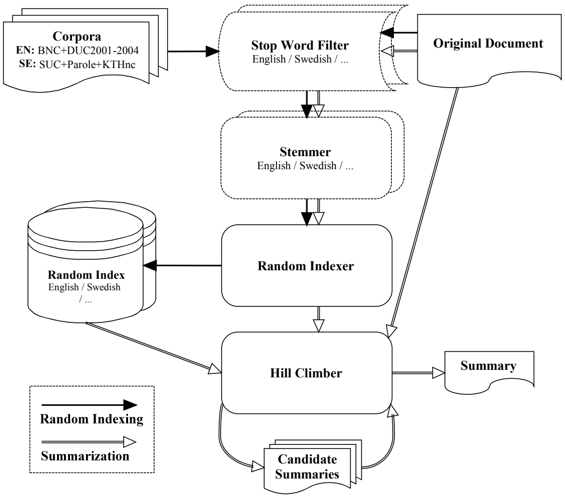


Figure 2: HolSum system layout. The candidate summaries are iteratively generated and evaluated (i.e. compared for semantic similarity against the original document).

Using a standard hill-climbing algorithm we then investigate all neighbors, looking for a better summary. The summaries that are defined as neighbors to a given summary are simply those that can be created by removing one sentence and adding another. Since sentences vary in length we also allow removing two sentences and adding one new, or just adding one new sentence. This allows for optimizing the summary size for the specified compression rate.

When all such summaries have been investigated, the one most similar to the original document is updated to be the currently best candidate and the process is repeated. If no other summary is better than the current candidate, the search is terminated. It is also possible to stop the search at any time if so desired, and return the best candidate so far. A schematic layout of the complete system can be found in Figure 2.

In our experiments on the texts provided for the Document Understanding Conferences (DUC 2007) the generated summaries are very short, about three sentences. This means that there are usually quite few, typically around five, search

```
Azerbaijani President Heydar Aliyev, who is considered the
most likely to win the presidential elections, cast his
vote today, Sunday, at one of the polling centers near
his residence in the center of the capital and took the
opportunity to attack his main opponent, Etibar Mammadov.
The president, who was elected in September 1993, said in a
statement to reporters that "one of the candidates, and you
know who I mean, asserts that he has a team and a program,
but when the country was on the verge of civil war in 1993,
Etibar Mammadov was involved in the political scene so why
did he not do anything and why did he not try to stop" the
tragedy.
```

Figure 3: Lead summary used as starting point for greedy search (ROUGE-1 37.8%, cosine 0.0310).

iterations. Some documents require quite many iterations before a local maximum is found, but these constitute a fairly small amount of the texts in the data set.

Example of a lead summary used as starting point for the greedy search can be found in Figure 3. As we can see, the lead summary is just the leading sentences within one document, and as such only covers the aspects of the document chosen to be presented there. Since our method tries to find a summary that is more similar to the view it has of the whole document, it thus transforms the initial summary into a summary with a wider coverage (if no slanting strategies are applied).

The local maximum summary, with a ROUGE-1 score of 44.0% and a 0.995 cosine closeness to the full document, reached from the lead summary given in Figure 3 is presented in Figure 4. Typically you will want as high ROUGE score as possible as this has been shown to correlate with summaries humans perceive as good summaries for a certain text (Hovy and Lin 2002, Lin and Hovy 2003a). The cosine angle between the summary vector and the document vector, both located in the same vector space, indicates the closeness, or likeness, between the current summary and the full document. This varies between -1 and 1 where 1 indicates complete similarity.

3.2.3 Evaluation

For reasons of comparability and the benefit of a human ceiling, we have chosen to mimic the evaluation set-up for task 2 in DUC 2004 (Over and Yen 2004). As in this evaluation campaign we have carried out our evaluation using ROUGEeval (Lin 2003) with the same data and model summaries. While our method itself

```
Supporters of Azerbaijani President Heydar Aliyev proclaimed
today, Monday, that he was reelected for a new term from
the first round that took place yesterday, Sunday, while
his main opponent Etibar Mammadov, declared that a second
round ought to be held.  The 4200 polling offices, under
the supervision of 180 observers from the Security and
Cooperation Organization in Europe, will remain open till
20:00 local time.  In order to win in the first round as
Aliyev hopes, a candidate must win more than 75% of the
votes with a turnout of over 25%.
```

Figure 4: Local maximum summary scoring ROUGE-1 44.0%, with a cosine similarity of 0.995.

is largely language independent, and thus should work comparably well on many other languages given enough raw text, the data prepared for the DUC evaluations is widely used and as such forms a basis for comparison with other systems and methods. The evaluation was carried out by first using all manually created 100 word summaries provided for DUC 2004 as reference summaries, trimming our system with different basic tokenizers and preprocessors (i.e. sentence splitting, stop word filtering and stemming), comparing our results to those reported in (Over and Yen 2004). Having reached a reasonable level of success we then compared against the complete set of man-made 100 word summaries from DUC 2001-2004 in order to verify our method on a larger test set.

The evaluation has been carried out by computing ROUGE scores on the system generated summaries using manual summaries provided for DUC as reference, or model summaries. The ROUGE score is a recall-based n-gram cooccurrence scoring metric that measures content similarity by computing the overlap of word n-grams occurring in both a system generated summary as well as a set of, usually man-made, model summaries. Throughout the evaluations we have, as in DUC 2004, used ROUGEeval-1.4.2 with the following settings:

```
rouge -a -c 95 -b 665 -m -n 4 -w 1.2
```

This means that we use a 95% confidence interval, truncate model and peer at 665 bytes, Porter Stem models and peers and calculate ROUGE-1..4. Also, stop words are not removed when calculating the score. ROUGE scores have in several studies been shown to correlate highly with human evaluation and has high recall and precision in predicting statistical significance of results comparing with its human counterpart (Lin and Hovy 2003b).

	DUC 2004	DUC 2001 - 2004
Human mean	42.6	39.7
Holistic-1000	34.1	32.4
Holistic-500	34.2	32.3
Holistic-250	33.9	32.0
Holistic-RAW	32.7	30.9
Holistic-noRI	30.3	28.5
Baseline-Lead	31.0	28.3

Table 1: ROUGE-1 scores, in %, for different dimensionality choices of the context vectors. RAW indicates no use of stemming and stop word filtering, and noRI uses a traditional $tf \cdot idf$ weighted vector space model instead of Random Indexing.

In our experiments ROUGE scores are in the case of DUC 2004 calculated over 114 system generated summaries, one for each document set, and in the case of DUC 2001-2004 over 291 summaries. A human ceiling (see Table 1) has for reference been calculated by, for each document set, taking the mean of the ROUGE scores for each man-made summary compared to the remaining man-made summaries (i.e. in turn treating each human-written summary as a system summary). On average there are about four man-made summaries available for each set. Also, we evaluate a baseline (lead), which is the initial sentences in each text up to the allowed summary length.

3.2.4 Results

In the evaluations here we have removed stop words and used stemming. Two brief evaluations not using these two strategies showed that both approaches result in considerable improvements, although even without the use of these techniques the system still improves on lead. We also evaluate the impact of the dimensionality chosen for the Random Indexing method by running our experiments for three different values for the dimensionality, building semantic representations using 250, 500 and 1000 dimensions. Our results show little variation over different dimensionalities. This means that as long as we do not choose too few dimensions, the dimensionality is not a parameter that needs considerable attention.

For each dimensionality we also calculated the mean performance using ten different random seeds, since there is a slight variation in how well the method works with different random projections. The dimensionality showing the most variation in our experiments spanned 33.8-34.4% ROUGE-1. Variations for the other dimensions were slightly less. As shown in Table 1, our best run resulted in a mean performance of 34.2%.

A ROUGE-1 score of about 34% on the DUC 2004 data set is not very impressive, but neither is it very bad. The best systems participating in the DUC 2004 evaluation campaign scored roughly 39% (Over and Yen 2004), with many systems scoring around 34% and some below. Concerning scores for ROUGE-2..4 our system unsurprisingly follows the pattern of the results reported in the DUC 2004 evaluation campaign, with considerably lower ROUGE-2 (mean 7.2% with 500 dimensions) and almost non-existing scores for ROUGE-3 (mean 2.3%) and ROUGE-4 (mean 1.0%).

Some naïve attempts at sentence compression by removing "uninteresting" text, such as removing anything mentioned within parenthesis were done. We also tried joining sentences together if the second sentence began with 'but', 'and', 'however', 'although' or similar text binding markers, indicating that the sentences were in some sense dependent. All such experiments, however, degraded the performance.

3.3 Trying Another Language: Swedish

Since the summarization method described above is relatively language independent, we decided to also evaluate it on Swedish. For this purpose we used the KTH Extract Corpus (Hassel and Dalianis 2005), a corpus of human produced extractive summaries of Swedish newspaper articles. These extracts were however not produced to give an overview of the whole contents of the texts, which our method attempts to do. The humans were instead more focused on finding the most important topic in the text and then providing mostly information relevant to that.

There are only 15 relatively short documents in this corpus. On average there are 20 human generated extracts for each document. These vary quite a lot in compression rate, even for a specific document. There are usually some sentences that are included in almost all extracts, though, so there is agreement on what the main topic is. In Figure 5 an example of the variation in selected sentences for one of the texts from the extract corpus is shown. As can be seen in this figure, the HolSum system tries to represent all parts of the text in the same proportion as in the source document. This is here illustrated by the system covering all three "information spikes", as chosen by the human informants.

As reference texts for the Random Indexing method we here used the Swedish Parole corpus (Gellerstam *et al.* 2000), 20 million words, the Stockholm-Umeå Corpus (Ejerhed *et al.* 1992), 1 million words, and the KTH News Corpus (Hassel

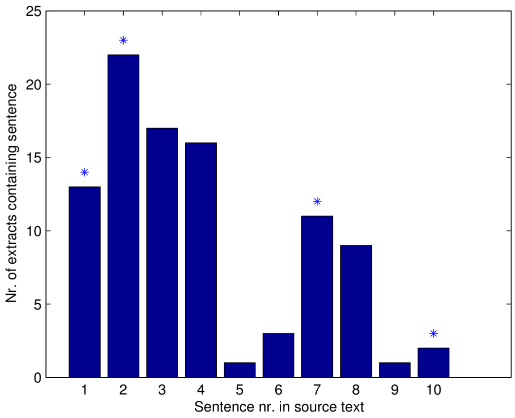


Figure 5: The number of human produced extracts that included each sentence from one of the Swedish corpus texts. There are a total of 27 human produced extracts for this text. This particular text contains 10 sentences, and sentences marked with a * are those selected by our system.

2001), 13 million words. We also used stemming and stop word filtering, since this worked well on the English texts.

3.3.1 Evaluation

When evaluating the Swedish summaries we calculated a weighted precision. The score for a sentence included in the summary is the number of human produced extracts that also included this sentence divided by the total number of human produced extracts for that text. The precision for the summary is then the average for all sentences in the summary.

A recall-like measurement was also calculated, since otherwise it would be best to simply pick a single sentence that the system is sure should be included. Each sentence that was included in at least one human produced extract, but not included in the summary to be evaluated, was also given a score as above, i.e. how often it was included by humans. The recall-like measurement is then the average score for all sentences not included in the summary but included in some human produced

	Included	Ignored	Perfect
Human	53	27	8
Lead, Short	55	29	2
Lead, Long	48	26	2
Random, Short	33	36	0.3
Random, Long	34	37	0
SweSum-above	53	28	3
SweSum-below	54	30	0
Holistic-500, Short	42	34	1
Holistic-500, Long	38	35	0

Table 2: Proportion of human produced extracts that included the sentences chosen by the system, in % (higher is better), and sentences ignored by the system but included by at least one human, also in % (lower is better). "Perfect" indicates for how many of the 15 documents a system generated an extract that was exactly the same as one of the human produced extracts.

extract. Sentences ignored by both the system and the humans have no impact in the evaluation.

Since the extracts vary so much in length we generated two different sets of summaries using our method. The first, called Holistic-long, was the summary most similar to the original text that was longer than the shortest human produced extract and shorter than the longest. This generally produced long summaries, since it is easier to achieve good coverage of the original text with many words than with few. Since long summaries will have lower precision we also generated summaries, called Holistic-short, that, while longer than the shortest human produced extract, were never longer than the average extract.

For both sets of summaries four different Random Indexes generated with four different seeds were used. The results in Table 2 are the mean values of these four sets. All values are within 1.5 percentage units of the mean value. We also compared our system to two baselines: *Lead*, the first sentences of the original text with a size as close to the system generated summary as possible; and *Random*, randomly chosen sentences up to the same size. We also calculated the agreement between the humans, by taking the average over all human produced extracts when treating them one at a time as a system generated summary instead.

Finally, we include figures for another summarization system, SweSum (Dalianis 2000, Hassel 2004), that has also been evaluated on this data set. SweSum uses both statistical and linguistic methods, as well as some heuristics, and its main domain is newspaper text. SweSum creates extracts, by scoring sentences for various criteria, then extracting high scoring sentences in the original text and joining them together. The sentence scores are calculated based on e.g. sentence position, occurrence of numerical data and highly frequent keywords. Two different sets of summaries were generated with SweSum, one with summaries strictly below the average human produced extract length and one with the shortest summary possible above the average length.

3.3.2 Results

As can be seen in Table 2, our system does not generate the same type of summaries as the others. Since our system tries to include the same proportions regarding different topics in the summary as was found in the original text, it has a quite low score with the precision-like measurement. This is natural, since the reference extracts normally only cover one topic. This also leads to a high (i.e. bad) score on the recall-like measurement, since the reference extracts include so much information regarding the main topic that our method discards some of it as redundant.

When generating shorter summaries the same sentences are of course still considered redundant by our method, so the recall-like figure is more or less unchanged. Since the extract is shorter, there is room for less information. This gives higher precision, since our method still agrees that the main topic should be covered, but now includes less information regarding other topics. As expected, it seems like using our method when single topic summaries is wanted does not give the best results. It can also be seen that outperforming the lead baseline on newspaper texts is very hard, since it performs on par with humans when generating shorter extracts. This means that this type of text is not very exciting to do summarization experiments on.

3.4 New Weighting Criteria: Keywords Come in Bursts

When constructing the semantic vector for a text, the context vector for each word is weighted with the importance of this word, by simply making the length of the vector proportional to the importance of the word. The weight could for instance be something simple, such as like in the previous sections making the length of the vector be $tf \cdot \log(idf)$, i.e. the term frequency and inverse document frequency. The term frequency is the frequency of the term within the given document and gives a measure of the importance of the term within that particular document. The inverse document frequency, on the other hand, is a measure of the general importance of the term – i.e. how specific the term is to said document (Salton and Buckley 1987).

In addition to the highly traditional $tf \cdot \log(idf)$ weighting scheme, we have also experimented with utilizing the “burstiness” of a word for term weighting.

	DUC 2004	DUC 2001 – 2004
Human	43	40
Burstyness, 1000	33.9	32.2
Burstyness, 500	33.7	32.1
Burstyness, 250	33.6	31.9
$tf \cdot \log(\mathit{idf})$, 1000	34.1	32.4
$tf \cdot \log(\mathit{idf})$, 500	34.2	32.3
$tf \cdot \log(\mathit{idf})$, 250	33.9	32.0
Baseline-Lead	31.0	28.3

Table 3: ROUGE-1 scores, in %, for burst weighting as well as the standard weighting criteria for reference. There are 114 documents from DUC 2004 and 291 from DUC 2001 – 2004.

Ortuño *et al.* (2002) have shown that the spatial information of a word, i.e. the way in which it is distributed in the text (independently of its relative frequency), is a good measure of the relevance of the word to the current text.

The burstiness of a word is here based on the standard deviation of the distance, in words, between different occurrences of this word in the text. Words that occur only with large distances between occurrences usually have a high standard deviation by chance, so the standard deviation is divided by the mean distance between occurrences. The final weight of a word is thus:

$$tf \cdot \frac{\sigma}{\mu}$$

where μ is the mean and σ the standard deviation of the distances between occurrences, in words.

3.4.1 Results

As before, we evaluated on three different dimensionality choices, 250, 500 and 1,000. Generally, as low dimensionality as possible is desirable, since processing times and memory usage is then lower. In Table 3 it can be seen that the variation between different dimensionalities is quite low. It is largest for $tf \cdot \log(\mathit{idf})$, where the mean value for dimensionality 250 is 32.0% and the mean value for 1,000 is 32.3% in the DUC 2001 – 2004 data set. This is nice, since it seems to be unimportant to spend a lot of time optimizing the choice of this parameter.

For each choice of dimensionality the mean performance using ten different random seeds was calculated. The impact of the randomness used in the method

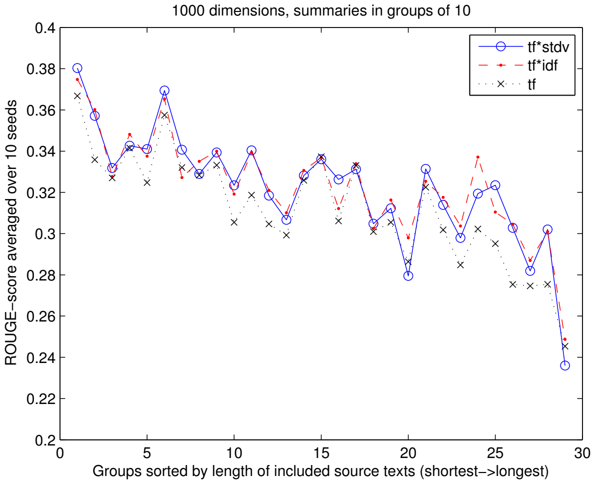


Figure 6: ROUGE-1 scores for three weighting schemes, divided into 29 groups of 10 summaries each sorted by compression rate. The leftmost group contains the summaries for the 10 shortest source texts while the rightmost group contains the summaries for the 10 longest.

seems larger than the impact of the dimensionality choice. The largest variation was for the dimensionality 500, spanning 33.1% – 34.3 % ROUGE-1 score in the DUC 2004 data set. Variations for the other dimensionalities were slightly less.

The choice between $tf \cdot \log(idf)$ or burstyness seems to have very little impact, the results are nearly identical in ROUGE-1 scores. This is further supported when plotting a graph, showing the ROUGE scores for three different weighting schemes. The first weighting scheme is $tf \cdot \log(idf)$, the second is burst weighting and the third is weighting only by the term frequency. In Figure 6 we can see that it is the term frequency that is pulling the most weight and that the inverse document frequency and the standard deviation seem to add roughly the same improvement.

It should, however, not come as such a surprise that the term frequency has the most impact during the accumulation of the context vectors. Since we apply stop word filtering prior to this step, we have already filtered out most of the highly frequent function words. This means that the remaining high frequent words are content words and as such good descriptors of the document being summarized.

In Figure 6 we can also see that summarizer performs best at low compressions rates. This is due to the fact that the more of the source text that is included in the summary, the higher the chance of selecting the same sentences, or choice of words, as the man-made summaries we are using as gold standard.

3.5 A New Search Strategy: Simulated Annealing

One obvious thought is that the greedy hill climbing might be a too simple search strategy and thus miss the best candidates available in the summary space. The best summaries may not lie down the path of always choosing the best neighbor. What if beyond one of the lesser neighbors lies an even better summary?

The method we used for investigating this idea is simulated annealing (Kirkpatrick *et al.* 1983), augmented with back-off heuristics. Instead of in each step choosing the best neighbor as our next transition point we may go to a randomly chosen neighbor, as long as it is better than the current summary. However, in doing this we also keep track of the best neighbor so far, and in the case that we venture to far down a slope[3] we can always go back to the best neighbor previously visited and start our search anew. A ban list containing all visited summaries, excluding the best summary so far, effectively hinders us from going down the same path again (not that it would have mattered much, bar computing time). This means that the annealing procedure will always perform at least on par with the greedy search regarding cosine scores.

With simulated annealing the cooling schedule is of great importance (Laarhoven and Aarts 1987). The cooling schedule is the factor that in each transition governs the probability of choosing a random better neighbor instead of the best neighbor. Two common formulas for calculating the cooling factor were used in these experiments. The first schedule was calculated using the following formula:

$$T_i = T_0 \left(\frac{T_N}{T_0}\right)^{\frac{i}{N}}$$

In this formula T_i is the probability of choosing a random better neighbor in step i, where i increases from 0 to $N = 100$ transitions. The initial probability T_0 is set to 100% and the lowest allowed probability to $T_N = 5\%$. This schedule starts with a high probability for random behavior and then rapidly reverts to a traditional greedy search. The second cooling schedule, using the same notation as above but

[3] In our case ten transitions without finding a new summary that is better than best one seen so far.

	DUC 2004	DUC 2001 – 2004
Human	43	40
Schedule 1, 1000	34.1	32.4
Schedule 1, 500	34.2	32.3
Schedule 1, 250	33.9	32.0
Schedule 2, 1000	34.2	32.4
Schedule 2, 500	34.2	32.3
Schedule 2, 250	34.0	32.0
Holistic-1000	34.1	32.4
Holistic-500	34.2	32.3
Holistic-250	33.9	32.0
Baseline-Lead	31.0	28.3

Table 4: ROUGE-1 scores, in %, for the the two annealing schedules as well as the standard greedy search for reference.

with T_N set to zero, was designed to revert to a greedy search more linearly:

$$T_i = T_0 - i\frac{T_0 - T_N}{N}$$

The algorithm was in both cases set to break when no known neighbors are better than the current summary and no previous state or neighbor has been better, in terms of cosine closeness, or the maximum number of 100 transitions has been reached. At this point the best state, current or previously visited, is returned. In most cases the maximum number of transitions was never reached.

3.5.1 Results

As can be seen in Table 4 the resulting summaries were in almost all cases identical to the summaries generated using the bare greedy search algorithm. In the as few as 7 cases out of 2910 where the summaries generated with a dimensionality of 500 differed, the second cooling schedule resulted in slightly higher ROUGE scores, but not enough to warrant the radically added computation time. For the same dimension the first schedule resulted in only one higher scoring summary.

Of course, a formula with a slower descent into a traditional greedy search could be used. However, this would probably lead to even further increased run times, depending on whether the cooling schedule in fact reaches a local optimum in fewer transitions or not. As it is, simulated annealing, using the two cooling

	DUC 2004	DUC 2001 – 2004
Human	43	40
rand, 1000	33.2	31.1
rand, 500	33.0	31.2
rand, 250	33.1	31.1
randlead, 1000	33.1	31.3
randlead, 500	33.2	31.3
randlead, 250	33.1	31.3
lead, 1000	34.1	32.4
lead, 500	34.2	32.3
lead, 250	33.9	32.0
Baseline-Lead	31.0	28.3

Table 5: ROUGE-1 scores, in %, for the two different random starting point strategies as well as the standard lead starting point for reference.

schedules presented here, in general takes about three times as long to generate the set of summaries evaluated in each run, compared to the standard greedy search.

3.6 Expanding the Search Scope: Different Points of Departure

Considering the approaches above, we have still only investigated a small fraction of the high-dimensional vector space representing all possible summaries. As previously stated it is simply not feasible to exhaustively search all possible summaries in pursuit of the best summary. Another option is to again put the greedy search to use, but this time giving it randomly chosen starting points. The idea here is that there may be better starting points than the leading sentences of the original text, thus taking other paths to possibly better summaries.

We have tried two approaches, where the first simply choses sentences randomly from the source text and concatenates them into an initial summary of desired length. The second, and slightly less naive approach, picks a random sentence in the source text and grabs it and the following couple of sentences to use as the initial summary for that text. After this the algorithm proceeds as before, transforming the initial summary until no better summary is found.

3.6.1 Results

One would like to believe that some difference in the results would show between these two approaches since the first obviously disregards any coherency in the text, while the other at least retains some. The second approach does however potentially breach coherency somewhat in that it may start e.g. in the middle of one paragraph and continue half-way into the next, or, when dealing with a concatenated set of topically related texts, come to span over a document boundary. However, as can be seen in Table 5, the results from both approaches are strikingly similar, giving further support to the notion that leading sentences of a document constitutes a stable starting point.

4 Conclusions

We have presented and evaluated an extraction based summarization method based on comparing whole summaries, not ranking individual extraction segments. It produces extracts that include the same proportions of topics as the original text. The method is largely language independent and requires no sophisticated tools, though stop word filtering and simple stemming was used in our experiments. For good performance, access to large amounts of raw text is needed, but for many languages this is readily available.

In the major part of our experiments we have used the leading sentences of a text as a starting point for our system since this itself usually constitutes a good summary. Though by doing this we limit our search for a better summary to a very limited area of the high-dimensional summary space. Since an exhaustive search of the vector space is not reasonable we have also sampled the space using some randomly chosen starting points, as well as used simulated annealing with the leading sentences as starting point. The results, however, show that using the lead summary as a starting point is a reliable heuristic also in this application.

Due to the fact that our method tries to cover all topics covered in the original text, it did not perform very well when evaluated against man-made extracts produced to cover mostly the main topic of a text. It did however perform well on short extracts derived from fairly long news texts when compared to man-made summaries, such as those used in the DUC 2004 summarization evaluation campaign. On this task the proposed method performs better than several of the systems evaluated there, but worse than the best systems.

Even though the HolSum summarizer does not outperform the best systems for English it is trivial to port to other languages. It also has the intuitively appealing property of optimizing semantic similarity between the generated summary and the text being summarized. Also, this property is not constrained to extractive summarization, even though we here use it to differentiate between extractive summaries. The summaries being evaluated and selected from could in practice be generated by any means, even being man-made.

References

Lou Burnard. 1995. The Users Reference Guide for the British National Corpus.

Hercules Dalianis. 2000. SweSum - A Text Summarizer for Swedish. Technical Report TRITA-NA-P0015, IPLab-174, KTH NADA, Sweden.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

DUC. 2007. Document Understanding Conferences. http://duc.nist.gov/.

Educational Testing Service. 2006. Test of English as a Foreign Language (TOEFL). http://www.ets.org/toefl.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. *SUC - The Stockholm-Umeå Corpus*, version 1.0 (suc 1.0). CD-ROM produced by the Dept of Linguistics, University of Stockholm and the Dept of Linguistics, University of Umeå. ISBN 91-7191-348-3.

Martin Gellerstam, Yvonne Cederholm, and Torgny Rasmark. 2000. The bank of Swedish. In *In the proceedings of Second International Conference on Language Resources and Evaluation. LREC-2000*, pages 329–333, Athens, Greece.

Zelig S Harris. 1968. *Mathematical Structures of Language*. New York: Wiley.

Martin Hassel. 2001. Internet as Corpus - Automatic Construction of a Swedish News Corpus. In *Proceedings of NODALIDA'01 - 13th Nordic Conference on Computational Linguistics*, Uppsala, Sweden, May 21-22 2001.

Martin Hassel. 2004. *Evaluation of Automatic Text Summarization - A practical implementation*. Licentiate thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden.

Martin Hassel. 2006. JavaSDM - A Java tool-kit for working with Random Indexing. http://www.nada.kth.se/∼xmartin/java/JavaSDM/.

Martin Hassel and Hercules Dalianis. 2005. Generation of Reference Summaries. In *Proceedings of 2nd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, Poznan, Poland, April 21-23 2005.

Eduard Hovy and Chin-Yew Lin. 2002. Manual and Automatic Evaluation of Summaries. In Udo Hahn and Donna Harman, editors, *Proceedings of the Workshop on Text Summarization at the 4Oth Meeting of the Association for Computational Linguistics*.

Scott Kirkpatrick, C. Daniel Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680.

Peter J. M. Laarhoven and Emile H. L. Aarts, editors. 1987. *Simulated annealing: theory and applications*. Kluwer Academic Publishers, Norwell, MA, USA. ISBN 9-027-72513-6.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.

Chin-Yew Lin. 2003. ROUGE: Recall-oriented understudy for gisting evaluation. http://www.isi.edu/∼cyl/ROUGE/.

Chin-Yew Lin and Eduard Hovy. 2003a. Automatic Evaluation of Summaries Using n-gram Co-occurrence Statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May 27 - June 1 2003.

Chin-Yew Lin and Eduard Hovy. 2003b. The potential and limitations of automatic sentence extraction for summarization. In Dragomir Radev and Simone Teufel, editors, *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, Edmonton, Alberta, Canada, May 31 - June 1 2003. Association for Computational Linguistics.

Kevin Lund, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the Cognitive Science Society*, pages 660–665, Hillsdale, N.J.: Erlbaum Publishers.

M. Ortuño, P. Carpena, P. Bernaola-Galvan, E. Munoz, and A. Somoza. 2002. Keyword detection in natural languages and DNA. *Europhysics Letters*, 57: 759–764.

Paul Over and James Yen. 2004. An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems. http://www-nlpir.nist.gov/projects/duc/pubs/2004slides/duc2004.intro.pdf.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference* *on Terminology and Knowledge Engineering, TKE 2005)*, Copenhagen, Denmark, August 16 2005.

Magnus Sahlgren. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Doctoral thesis, Department of Linguistics, Stockholm University, Stockholm, Sweden.

Magnus Sahlgren and Rickard Cöster. 2004. Using Bag-of-Concepts to Improve the Performance of Support Vector Machines in Text Categorization. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004*, Geneva, Switzerland, August 23-27 2004.

Gerard Salton and Chris Buckley. 1987. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

Klaus Zechner. 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *The 16th International Conference on Computational Linguistics, COLING 1996*, pages 986–989, Center for Sprogteknologi, Copenhagen, Denmark, August 5-9 1996.