Evaluation of Text Summarization in a Cross-lingual Information Retrieval Framework

Summer 2001 Johns Hopkins Workshop
Final Report

Dragomir Radev, Simone Teufel, Horacio Saggion, Wai Lam
John Blitzer, Arda Çelebi, Hong Qi, Elliott Drabek, and Danyu Liu

Contents

List of Figures

Abstract

We report on research in multi-document summarization and on evaluation of summarization in the framework of cross-lingual information retrieval. This work was carried out during a summer workshop on Language Engineering held at Johns Hopkins University by a team of nine researchers from seven universities. The goals of the research were as follows: (1) to develop a toolkit for evaluation of single-document and multi-document summarizers, (2) to develop a modular multi-document summarizer, called MEAD, that works in both English and Chinese, and (3) to perform a meta-evaluation of four automatic summarizers, including MEAD, using several types of evaluation measures: some currently used by summarization researchers and a couple of novel techniques.

Central to the experiments in this workshop was the cross-lingual experimental setup based on a large-scale Chinese and English parallel corpus. An extensive set of human judgments were specifically prepared by the Linguistic Data Consortium for our research. These human judgments include a) which documents are relevant to a certain query and b) which sentences in the relevant documents are most relevant to the query and which therefore constitute a good summary of the cluster. These judgments were used to construct variable-length multi- and single document summaries as model summaries. Since one of the novel evaluation metrics that we used, *Relevance Correlation*, is based on the premise that good summaries preserve query relevance both within a language and across languages, we made use of a cross-lingual Information Retrieval (IR) engine.

We evaluated the quality of the automatic summaries using *co-selection* and *content-based evaluation*, two established techniques. A relatively new metric, *relative utility*, was also extensively tested. Part of the new scientific contribution is the measurement of relevance correlation, which we introduced and systematically examined in this workshop. Relevance correlation measures the quality of summaries in comparison to the entire documents as a function of how much document relevance drops if *summaries* are indexed instead of documents. Our results show that this measure is sensible, in that it correlates with more established evaluation measures.

Another contribution is the cross-lingual setup which allows us to automatically translate English queries into Chinese, perform Chinese IR with or without summarization. This allows us to calculate relevance correlation for English and for Chinese in parallel (i.e., for the same queries) and to take direct cross-lingual comparisons of evaluations. Additionally, an alternative way of constructing Chinese model summaries from English ones was implemented which relies on the sentence alignment of English and Chinese documents.

The results of our large-scale meta-evaluation are numerous, but some of the highlights are the following: (1) All evaluation measures rank human summaries first, which is an appropriate and expected property of such measures, (2) Both relevance correlation and the content-based measures place leading sentence extracts ahead of the more sophisticated summarizers, (3) Relative utility ranks our system, MEAD, as the best summarizer for shorter summaries, although for longer summaries, lead-based summaries outperform MEAD, (4) Co-selection measurements show overall low agreement amongst humans (above chance), whereas relative utility reports higher numbers on the same data (but does not normalize for chance).

The deliverable resources and software include: (1) a turn-key extractive multi-document summarizer, MEAD, which allows users to add their own features based on single sentences or pairs of sentences, (2) a large corpus of summaries produced by several automatic methods, including baseline and random summaries, (3) a collection of manual summaries produced by the Linguistic Data Consortium (LDC), (4) a battery of evaluation routines, (5) a collection of IR queries in English and Chinese and the corresponding relevance judgments from the Hong Kong news collection, (6) SMART relevance outputs for both full documents and summaries, (7) XML tools for processing of documents and summaries.

Chapter 1

Introduction

Given the enormous amount of textual information on the Internet, one worthy goal of research in Natural Language Processing and Information Retrieval is to develop techniques for automatic text summarization. A team of researchers gathered at Johns Hopkins University in Summer 2001 to address the following goals: (1) to develop a modular multi-document summarizer that achieves state-of-the art performance in both English and Chinese, (2) to develop a toolkit for evaluation of both single-document and multi-document summarizers, and (3) to perform a meta-evaluation of six summarizers using four classes of evaluation measures: co-selection, content-based, relative utility, and relevance correlation. All three goals were successfully met. The current distribution of the MEAD system includes (1) a turn-key extractive multi-document summarizer, (2) a large corpus of summaries produced by different methods, including baseline and random summaries, (3) a collection of manual summaries (produced by LDC, the Linguistic Data Consortium), (4) a battery of evaluation routines, (5) a collection of IR queries in English and Chinese and the corresponding relevance judgments from the Hong Kong news collection, (6) SMART relevance outputs for both full documents and summaries, (7) XML tools for manipulation of documents and summaries.

In this report we describe the MEAD project in detail and specifically, the summarizer itself, the corpus that we prepared and annotated, as well as a new evaluation metric for summary evaluation, *Relevance Correlation*. We present a comparison of MEAD with several other summarizers as well as a meta-evaluation comparing eight evaluation metrics: Precision/Recall, Percent Agreement, Kappa, Relative Utility, Relevance Correlation, and three types of content-based measures (cosine, longest common subsequence, and word overlap).

1.1 The Johns Hopkins workshop

The Summer workshop on Language and Speech processing has been held at Johns Hopkins University since 1996. Each year, a number of projects (usually four) are selected from a number of proposals. Other projects in recent years have included Statistical Machine Translation, Novelty Detection, and Graphical Models for Speech Processing.

1.2 Research hypotheses

We tried to address the following research hypotheses:

1. Good summaries preserve relevance: in other words, if an information retrieval system is used to rank for relevance to a given query (a) a set of documents and (b) a set of the corresponding summaries of these documents, the rankings will be quite similar. We call such an evaluation *relevance correlation*.

2. Good query translation preserves relevance: if documents are first translated into a different language and then summarized, there will still be a correlation between the relevance rankings for the documents in the two languages.

3. Humans agree on sentence utility: we ask human judges to specify how central a sentence is to a cluster of related documents, we call this type of relevance *sentence utility*. Our hypothesis is that the utility given to a sentence by different judges will be similar.

4. Relevance correlation correlates with established methods for summary evaluation

1.3 Technical objectives

At the beginning of the workshop, we set the following technical objectives:

1. To develop a summarization evaluation toolkit allowing for meta-evaluation: extractive and non-extractive

2. To develop a summarization toolkit including a modular state-of-the-art summarizer: single/multi document, generic/query-based, English/Chinese

3. To produce an annotated corpus for further research in text summarization

After completion of these tasks, our plan is for users to be able to perform the following activities:

1. Evaluate an existing summarizer.

2. Build a summarizer from scratch.

3. Test a summarization feature.

4. Test a new evaluation metric.

5. Test a query translation system.

Research in text summarization is traditionally constrained by the following problems: limited resources for training, lack of standard testbeds that can be used to compare different summarizers, and no clear understanding of the correlation between different summarization methods.

1.4 Participants

The work presented in this report was carried out mainly at the Johns Hopkins summer workshop in 2001 although a large portion was done both before and after the workshop itself. The workshop team includes the following people:

- Dragomir Radev received his PhD in Computer Science from Columbia University. He worked at IBM's TJ Watson Research Center in Hawthorne, NY before coming to the University of Michigan where he is currently Assistant Professor of Information, of Electrical Engineering and Computer Science, and of Linguistics. He is mainly interested in natural language processing and information retrieval. He heads the Computational Linguistics And Information Retrieval group (CLAIR) at Michigan. His most recent projects are on multi-document summarization, cross-document structure theory, and question answering.

- Simone Teufel received her PhD from Edinburgh University in 1999, and her first degree in Computer Science and Computational Linguistics from the University of Stuttgart in 1994. Until 1995, she worked on POS-Tagging of German, lexicon building and syntactic annotation schemes. Her graduate thesis is a summarization system, based on a multidisciplinary study of the summarization of scientific articles, and on the exploitation of particular speech acts found in scientific articles. As a postdoctoral researcher at Columbia University between 2000 and 2001, she worked on term identification, IR and summarization in the medical domain. She is now a lecturer in the Computer Laboratory, Cambridge University, UK.

- Wai Lam received a Ph.D. in Computer Science from the University of Waterloo, Canada in 1994. He worked as a visiting Research Associate at Indiana University Purdue University Indianapolis and as a Postdoctoral Fellow in University of Iowa. He joined the Department of Systems Engineering and Engineering Management in the Chinese University of Hong Kong in 1996 as Assistant Professor. In August 2001, he became Associate Professor. His current interests include intelligent information retrieval, text mining, machine learning, reasoning under uncertainty, and digital libraries.
- Horacio Saggion received his PhD from Université de Montréal, Canada, in 2000, and his Master degree from the University of Campinas, UNICAMP, Brazil, in 1995. He studied Computer Science in the Computer Science Department at Universidad de Buenos Aires, Argentina. He worked many years as teaching assistant and research assistant at the Computer Science Department and as System Programmer for the industry. He is currently research assistant in the Natural Language Processing group at the Department of Computer Science, University of Sheffield, UK, where he is involved in two projects on Information Extraction and Multimedia Summarization: "The Multimedia Indexing and Searching Environment" and "The Scene of Crime Information System." He is mainly involved in the use of symbolic techniques for NLP, nevertheless he believes that robust and practical solutions to many problems in NLP should be developed with a wise combination of statistical and symbolic knowledge and techniques. His main interests in NLP are text summarization, shallow natural language processing, text structure, discourse interpretation, and natural language generation.
- John Blitzer is a senior at Cornell University, Ithaca, NY. In Fall 2002, he will be a PhD student at the University of Pennsylvania.
- Arda Çelebi is a senior at Bilkent University, Ankara, Turkey. In Fall 2002, he will be a PhD student at the University of Southern California's Information Sciences Institute.
- Elliott Drabek is a PhD student at Johns Hopkins University.
- Danyu Liu is a PhD student at the University of Alabama.
- Hong Qi is a PhD student at the University of Michigan, Ann Arbor.

1.5 Plan of the report

This report includes seven sections and one appendix. The next chapter describes the framework in which the experiments were performed. Chapters 3 and 4 present an overview of the methods used for information retrieval and for summarization, respectively. The following chapter (Chapter 5) describes the techniques used to compare the different summarizers. Chapter 6 presents our results, grouped by evaluation method, while Chapter 7 concludes the report. The user documentation associated with the MEAD summarizer is included in Appendix A.

Chapter 2

Experimental Framework

2.1 Overview

2.1.1 Research questions

As motivated in the previous chapter, we set out to answer the following questions:

- Which summarizer, out of a set of automatic summarizers, creates extracts that are most *similar* to extracts a human would have created?
- How does this summary performance relate to certain well-known baselines?

These questions can be answered in the mono-lingual case, as soon as a reasonable number of human extracts are available. We answer these questions, but our setup is more sophisticated. Because we operate in a cross-lingual IR framework, we also ask the following questions:

- How well does the IR engine work for the language in which the queries are written (in our case, English)?
- How well does it perform if the queries are translated automatically into the parallel language (in our case, Chinese)?
- How much worse is this compared to the case where the translation is done *manually*?

Due to the unique setup created in this workshop, where we have at our disposal a large-scale, parallel English–Chinese newspaper corpus with IR relevance judgements *and* judgements about how relevant single sentences contained in the documents are to a query, we can also answer more complicated questions such as:

- How much does IR performance decrease if we index summaries instead of the entire document? In this case we restrict the information available to the IR engine to the supposedly more "important" parts of the document which the summarisers have identified for us.
- How does this new measurement, which we call *relevance correlation*, relate to the more established summary quality metrics of *similarity* mentioned above?

It is also a feature of our research that we address both multi-document and single-document summarization. One last point we address is the difference of extracts and manual summaries, i.e. summaries written by a human from scratch. We obtained manual summaries written by 3 human judges from the LDC, who summarized sets of 10 documents in 50, 100, and 200 words. We believe that such a resource is very valuable, as the highest-quality automatic summaries of the future will probably mirror more and more human summaries, and move away from sentence extracts. The final research question we address is:

- Using the evaluation metrics available to compare non-verbatim text, how similar are human extracts and human summaries, and how similar are automatic extracts to either of these (human extracts and summaries)?

These sets of questions lead to the exciting, new experimental setup for a meta-evaluation which we performed in the workshop. This chapter explains the entire meta-evaluation framework we employed: the data we used, the human annotation collected, the corpus processing and automatic summarization used, and the definition of baselines.

2.1.2 Overview of setup

The project setup includes a multilingual IR system and several sentence extractors for the single-document case. We compare the output of an IR system with the output of the human relevance judgement. Our setup also includes a sentence extraction step, performed by different summarizers. The IR system output, using an index created from the summaries, was compared to system performance when indexing took place on the entire document. IR performance was measured by comparison to the relevance judgements.

Similarity of extracts to human extracts was measured in parallel, using a different set of judgements created by our judges, which we called 'sentence judgements'. These were created by asking the judges how relevant each sentence in 10 of the relevant documents was to the query.

The setup in the cross-lingual case assumes that the corpus is parallel, i.e. that each English document can be aligned with a Chinese document, which is a translation of the document. This is given in our case. As we did not have Chinese judgements at our disposal who could have performed Chinese relevance and sentence judgements (and in order to keep annotation cost down), we operate under certain assumptions to duplicate Chinese "judgements" from the English ones:

- If an English document is relevant to a query, so is its translation into Chinese.
- If an English *sentence* is relevant to a query, so is its translation into Chinese.

Note that the last step requires us to find alignments of English and Chinese sentences; this is a well-known alignment problem which we implemented in the course of the workshop.

Our sentence extractors run on Chinese and on English text. This fact creates interesting possibilities for evaluation, as Chinese extracts by alignment can then be compared to the Chinese extracts generated by running the sentence extractors directly on Chinese text.

2.2 Corpus and automatic corpus processing

We use a parallel corpus of English and Chinese (Cantonese) texts which are translations or near translations of each other. The corpus consists of 18,146 document pairs covering 1997–2000. The corpus, called the *Hong Kong Newspaper Corpus* (corpus number LDC2000T46), is provided by the Linguistic Data Consortium (LDC). The texts are not typical news articles. The Hong Kong Newspaper mainly publishes announcements of the local administration and descriptions of municipal events, such as an anniversary of the fire department, or seasonal festivals.

The average size in words for a document is 347.8 for English and 325.2 for Chinese, in sentences it is 16.2 and 15.5, respectively.

Each document in the corpus was further automatically processed in order to add structural and linguistic information, cf. the overview in figure 2.2. The annotation for each document includes information about the document identity, its language and its translation. Plain text in English and Chinese was processed in order to identify the main title and the text of the news article. For the purpose of our research we also needed to split the corpus into sentences and words. English documents were annotated with parts of speech and morphologic information. Both Chinese and English text was annotated with Named Entity tags, and an algorithm for alignment of English and Chinese sentences was implemented.

We used the following naming convention: File names are of the form `yyyymmdd_nnn.[ce]` where `yyyy` is the year, `mm` is the month, `dd` is the day, and `nnn` is a sequential number.

All corpus information is encoded in XML, and a number of DTDs (document type descriptions) were written to describe the structure of the document after each processing step. DTDs give the logical structure of an XML file in Backus-Naur form. The DTDs created for this project are given in Appendix A.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE DOCSENT SYSTEM "/export/ws01summ/dtd/docsent.dtd" >
<DOCSENT DID='D-19980303_004.e' DOCNO='2203' LANG='ENG' CORR-DOC='D-
19980303_004.c'>
<BODY>
<HEADLINE><S PAR="1" RSNT="1" SNO="1"> Joseph W P Wong accepts ATV's apol-
ogy </S></HEADLINE>
<TEXT>
<S PAR='2' RSNT='1' SNO='2'>The Secretary for Education and Manpower, Mr
Joseph W P Wong, said today (Tuesday) that he had accepted the apology of
Asia Television Limited (ATV) over the remarks made on him in the ATV pro-
gramme "Hong Kong Affairs" last Monday (February 23) and would not pursue
the matter further.</S>
</TEXT>
</BODY>
</DOCSENT>
```

Figure 2.1: Document 19980303_004.e annotated with sentence boundaries.

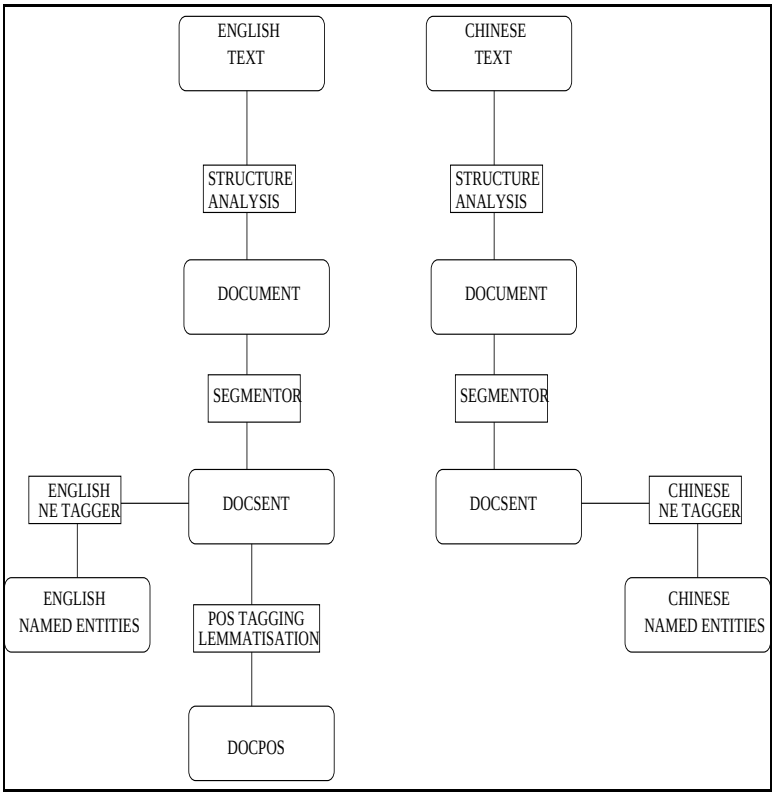


Figure 2.2: Linguistic Processing of the HK corpus

Using XML proved a software engineering advantage in the project, as many modules had to be interfaced by several programmers. XML validation made it very easy to check for errors in the input and output of each module in the pipeline.

2.2.1 Linguistic processing (English)

The first line is considered a headline and all the lines from the second to the end of the file are considered the text of the article. We removed "end/" statements and other non-textual entities occurring in the corpus. The output of

this process is stored in files conforming the *document* DTD specification (cf. Appendix A).

In order to further process English texts in *document* format, we use a pipelined implementation based on LTG's TTT (Text Tokenization Tool (Grover et al., 1999)), a modular package which includes a symbolic tokenizer written in a regular language, a statistical sentence segmenter (Mikheev, 2000) and a statistical part-of-speech (POS) tagger distributed with TTT. The POS tags used are those from the Penn Tagset. The tagger comes trained on 4 million words of the Wall Street Journal. Semi-automatic corrections of sentence boundaries had to be made in those sets of documents where human sentence segmentation was available, as there could not be any conflicts in sentence boundaries between our automatic sentence extractors and the human target summaries.

The output of the statistical sentence segmenter is stored in files conforming the *docsent* DTD specification, cf. figure 2.1. The text in this format is used by all summarization algorithms.

We used the lemmatiser developed at the University of Sheffield (Humphreys et al., 2000) and used in different natural language processing systems. The lemmatiser produces a lemma form for all nouns and verbs in the document. The work is accomplished by using a set of regular expressions and a list of exceptions are used for analysis: for example the form "expresses" matches the regular pattern "ANY+ DOUBLE "ES"", producing the root "express" and the affix "s". The list of exceptions was derived from WordNet (Fellbaum, 1998) and other corpora making the module domain independent. The input to the lemmatiser is a tagged list of nouns and verbs, the output consists of a lemma and suffix (possibly null) for each unit. Additionally, all lemmas are stored in lowercase. In Figure 2.2.1 we show a sentence from the corpus after lemmatisation. The output of preprocessing is stored in files conforming the *docpos* dtd specification. The information in these files is used during evaluation of content based similarity measures.


```
<S PAR='3' RSNT='2' SNO='4'><W C='DT' L='these'>These</W><W C='JJ' L='foreign'>foreign</W><W
C='NN' L='currency'>currency</W><W C='NNS' L='asset'>assets</W><W C='VBP' L='be'>are</W><W
C='IN' L='among'>among</W><W C='DT' L='the'>the</W><W C='NN' L='world'>world</W><W
C='POS' L="'s">'s</W><W C='JJS' L='largest'>largest</W><W C='JJ' L='such'>such</W><W C='NN'
L='holding'>holdings</W><W C='.' L='.'>.</W></S>
```


Figure 2.3: Sentence after tokenization, tagging, and lemmatization

2.2.2 Linguistic processing (Chinese)

Sentence segmentation in Chinese is based on punctuation. Unlike English, where punctuation can be part of words (cf. *"Dr."*), this is not the case in Chinese. We constructed a list of punctuation symbols that usually indicate the end of sentences. Then we used a greedy search to find the longest match of these punctuations.

Another processing required for Chinese texts is word segmentation. In a piece of Chinese text, there is no word delimiter between Chinese characters. The objective of word segmentation is to locate meaning words in Chinese texts. To do this, we make use of a Chinese word lexicon. A maximal matching algorithm is employed to match the longest possible word in the lexicon. The code for word segmentation was originally obtained from a Perl package located at (http://www.mandarintools.com/segmenter.html). We adapted this package from GB to BIG5 encoding in order to process Chinese. In Figure 2.4, we show a small Chinese document annotated with sentence boundaries.

2.2.3 Named entity detection in English and Chinese

It has been shown that the performance of text summarization systems can be improved by Named Entity Detection (Aone et al., 1999), so we decided to include this information into our summarizer. Named Entity (NE) Detection is the process of identifying and categorising names in texts. For instance, in the Message Understanding Conferences (MUC) (Grishman and Sundheim, 1996), the NE detection task consisted of the identification of seven types of NE: PERSON, ORGANIZATION, LOCATION, DATE, TIME, MONEY and PERCENT.

We use IdentiFinder (BBN, 2000), a probabilistic natural language software tool that scans text to locate NEs. The tool analyzes training data, counts and compiles statistics about the training data, convert those statistics into probabilistic models, applies those models to the NE task and outputs the same text with SGM marked-up text. The software is available in both English and Chinese; we used it used with the pre-trained models. An example of the NE detection task in English is shown in Figure 2.5.

```
<?xml version="1.0"?>
<!DOCTYPE DOCSENT SYSTEM "/export/ws01summ/dtd
/docsent.dtd" >
<DOCSENT DID="D-19980303_004.c" DOCNO="2203"
LANG="CHIN" CORR-DOC="D-19980303_004.e">
<BODY>
<HEADLINE>
<S PAR="1" RSNT="1" SNO="1">   王永平 接納 亞
洲 電視 道歉 </S>
</HEADLINE>
<TEXT>
<S PAR="2" RSNT="1" SNO="2">教育 統籌 局 局長
王永平 今日 （ 星期二 ） 表示 ， 他 已 接納 亞
洲 電視 有限 公司 對 其 在 上 星期一 （ 二月
二十三 日 ） 「 港 是 港 非 」 節目 發表 的 評
論 所 作出 的 道歉 ， 並 對 今次 事件 ， 將 不
再 跟進 。</S>
</TEXT>
</BODY>
</DOCSENT>
```

CNPRINT THU 09-MAY-02 16:25:47 HKT 1.

Figure 2.4: Chinese document 19980303_004.c

<S PAR=’2’ RSNT=’1’ SNO=’2’>Gross Domestic Product (GDP) grew by <NUMEX TYPE=”PERCENT”>6.4%</NUMEX> in real terms in the <TIMEX TYPE=”DATE”>second quarter of 1997</TIMEX> over a year earlier, further up from the <NUMEX TYPE=”PERCENT”>6.1%</NUMEX> increase in the <TIMEX TYPE=”DATE”>first quarter</TIMEX>.</S> <S PAR=’3’ RSNT=’1’ SNO=’3’>These are shown in the preliminary estimates of the expenditure-based GDP for the <TIMEX TYPE=”DATE”>second quarter of 1997</TIMEX> and revised estimates for earlier periods released today (<TIMEX TYPE=”DATE”>Monday</TIMEX>) by the <ENAMEX TYPE=”ORGANIZATION”>Census and Statistics Department</ENAMEX>.</S>

Figure 2.5: NEs in the English Corpus

2.2.4 English-Chinese sentence alignment

We identified correspondences between English sentences and the translations of the sentences in the respective Chinese document. This problem is not trivial, as the correspondence is not always 1:1. Translation is not trivial: sentences might be dropped, or two short sentences might be translated by one long one.

We use Church and Gale’s (1990) algorithm for alignment. It is based on a very simple statistical model of character lengths. The basic assumption is that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences. A probabilistic score is assigned to each pair of proposed sentence pairs, based on the ratio of lengths of the two sentences (in characters) and the variance of this ratio. The probabilistic score is used in a dynamic programming framework in order to find the maximum likelihood alignment of sentence.

The correspondences we model is 0:1, 1:0, 1:1, 1:2, 2:1 and 2:2. The distribution of each kind of correspondence is shown in figure 2.6.

The information about sentence alignment is kept in tables and is used in the cross-lingual evaluation.

Alignment Type	No of Pairs	(Percent)
$0 \Rightarrow 1$	357	(0.13%)
$1 \Rightarrow 0$	751	(0.28%)
$1 \Rightarrow 1$	215,296	(81.65%)
$1 \Rightarrow 2$	17,412	(6.60%)
$2 \Rightarrow 1$	29,056	(11.04%)
$2 \Rightarrow 2$	801	(0.30%)
Total	312,544	

Figure 2.6: Alignment figures

Queries 1–10	
Group 125	Narcotics Rehabilitation
Group 241	Fire safety, building management concerns
Group 323	Battle against disc piracy
Group 551	Natural disaster victims aided
Group 112	Autumn and sports carnivals
Group 199	Intellectual Property Rights
Group 398	Flu results in Health Controls
Group 883	Public health concerns cause food-business closings
Group 1014	Traffic Safety Enforcement
Group 1197	Museums: exhibits/hours
Queries 11–20	
Group 447	Housing (Amendment) Bill Brings Assorted Improvements
Group 827	Health education for youngsters
Group 885	Customs combats contraband/dutiable cigarette operations
Group 2	Meetings with foreign leaders
Group 46	Improving Employment Opportunities
Group 54	Illegal immigrants
Group 60	Customs staff doing good job.
Group 61	Permits for charitable fund raising
Group 62	Y2K readiness
Group 1018	Flower shows

Figure 2.7: 20 queries produced by the LDC (development corpus)

2.3 Human annotation

2.3.1 Queries and Clusters

LDC annotators developed 40 queries that cover a variety of subjects such as “narcotics rehabilitation” (the first 20 are shown in figure 2.7). Using an in-house information retrieval engine and human revision, the judges obtained documents highly relevant to the queries. The 10 most relevant (according to human assessors) were used to construct “clusters”. These 40 clusters of documents were used during the workshop for training and some specific evaluations.

In our workshop, Chinese translations of each query were produced by native speakers (workshop participants).

Figure 2.8 shows how we represent a query. Figure 2.9 shows the contents of cluster 125. The document IDs correspond to the HKNews corpus and indicate the year, month, day, and story number for each document.

```
<!ELEMENT QUERY (TITLE,DESCRIPTION?,NARRATIVE?)>
<!ATTLIST QUERY
   QID  CDATA #REQUIRED
   QNO  CDATA #REQUIRED
   LANG (CHIN|ENG) "ENG"
   TRANSLATED (YES|NO) "NO"
   ORIGLANG (CHIN|ENG) "CHIN"
   TRANS-METHOD (AUTO|MAN) "AUTO">

<!-- QID: unique query no, eg. 125-CA or 125-E
     QNO: LDC query no for content, eg. 125
     LANG: of query
     TRANSLATED: is it an original query or not?
     ORIGLANG: If translated, from which language (from the other
     one, of course
!)
     TRANS-METHOD: Automatically translated or manually? -->

<!ELEMENT TITLE       (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT NARRATIVE   (#PCDATA)>
```

Figure 2.8: Sample query

```
<?xml version='1.0'?>
<CLUSTER LANG="ENG">
        <D DID="D-20000408_011.e" />
        <D DID="D-19990927_011.e" />
        <D DID="D-19990425_009.e" />
        <D DID="D-19990218_009.e" />
        <D DID="D-19990829_012.e" />
        <D DID="D-19990729_008.e" />
        <D DID="D-19980430_016.e" />
        <D DID="D-19990211_009.e" />
        <D DID="D-19980306_007.e" />
        <D DID="D-19990802_006.e" />
</CLUSTER>
```

Figure 2.9: Sample cluster

2.3.2 Sentence judgements

We also asked the human judges to produce sentence relevance judgements. Three human annotators from LDC judged each sentence within the 10 relevant documents in each cluster for relevance to the query. They assigned each sentence a score on a scale from 0 to 10, expressing the importance of this sentence in the summary (Radev et al., 2000). This type of annotation is called “utility judgement”.

All sentence utility scores given by the judges for a given cluster are represented in a document complying to the *sentjudge* DTD, an example of which is shown in Figure 2.10.

2.3.3 Target summaries

The sentence judgements annotation discussed in the previous section allows us to compile human-generated ’ideal’ summaries at different compression rates, which is one gold-standard we use for our different measures of sentence-based agreement, both between the human agreement and between the system and the human annotators. We call this gold standard “human extracts”.

DOC:SENT	JUDGE1	JUDGE2	JUDGE3	TOTAL
19980306_007:1	4	6	9	19
19980306_007:2	5	10	9	24
19980306_007:3	4	9	7	20
19980306_007:4	4	9	8	21
19980306_007:5	5	8	8	21
19980306_007:6	4	9	5	18
19980306_007:7	4	9	6	19
19980306_007:8	5	7	8	20
...				
20000408_011:13	1	5	3	9
20000408_011:14	6	4	2	12
20000408_011:15	2	6	6	14
...				

Figure 2.10: Sentjudge: sentence utilities as assigned by the judges - cluster 125

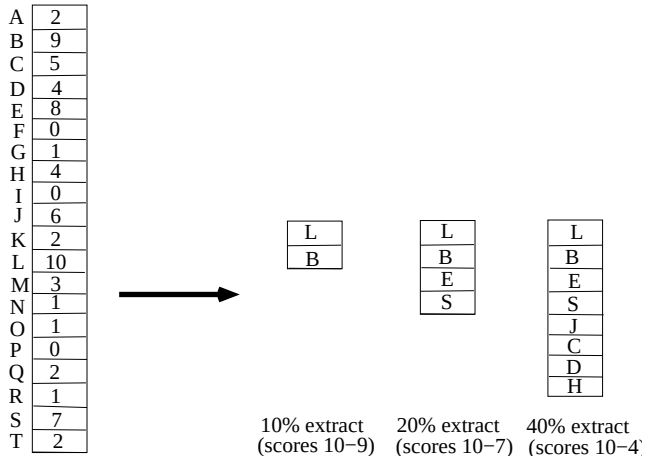


Figure 2.11: Creation of target summaries from sentence judgements

For each compression rate (5, 10, 20, 30, and 40%), the top n% ranks of utility judgements were taken to make up the target summary, as figure 2.11 shows. Because of limited time, we were only able to evaluate summaries produced at the low lengths of 5, 10, 20, 30, and 40%. For the purposes of evaluation and meta-evaluation discussed in this paper, these lengths were more than adequate, but future work may involve summaries of higher length.

Summary length and summarization policy

We measured the length of a summary in two different ways: by words and by sentences. When measuring the length of summary by sentences, for a document of n sentences at a rate of p%, we produce a summary of $\lceil \frac{n*p}{100} \rceil$ sentences.

Consider figure 2.10 again. The total number of sentences in cluster 125 is 232. By convention, a 10% summary will contain 24 sentences (23.2 rounded up).

When measuring summary length in words, we couldn't expect the exact results given to us by sentence-based compression, and since different summarizers extracted sentences of very different lengths, we decided on the following algorithm:
For a document of n words at a length of p%, let $\$idealsummary = \lceil \frac{n*p}{100} \rceil$.

```
while # of words in summary < $idealsummary{
```

```
  Add a sentence to the summary;
}

if (# of words in summary > $idealsummary + $idealsummary * 0.1){
  subtract the last sentence from the summary;
  if (# of words in summary < $idealsummary - $idealsummary * 0.1){
    add the last sentence to the summary if $random > 0.5;
  }
}
```

2.4 Experimental setup

2.4.1 Single-document case

Figure 2.12 shows part of the evaluations set up for single document extracts. The input to the process is a query and the corpus, i.e. the entire collection of documents. The IR engine SMART then outputs a ranked list of documents.

The first IR evaluation measure is measured by comparing this list to the lists created by humans mentioned earlier ("IR results"). To evaluate the quality of a certain summarizer we compare the IR output achieved this way to the IR output achieved on the full documents. This is the new measure introduced in this workshop, called Relevance Correlation (cf. "Correlation" in Figure 2.12).

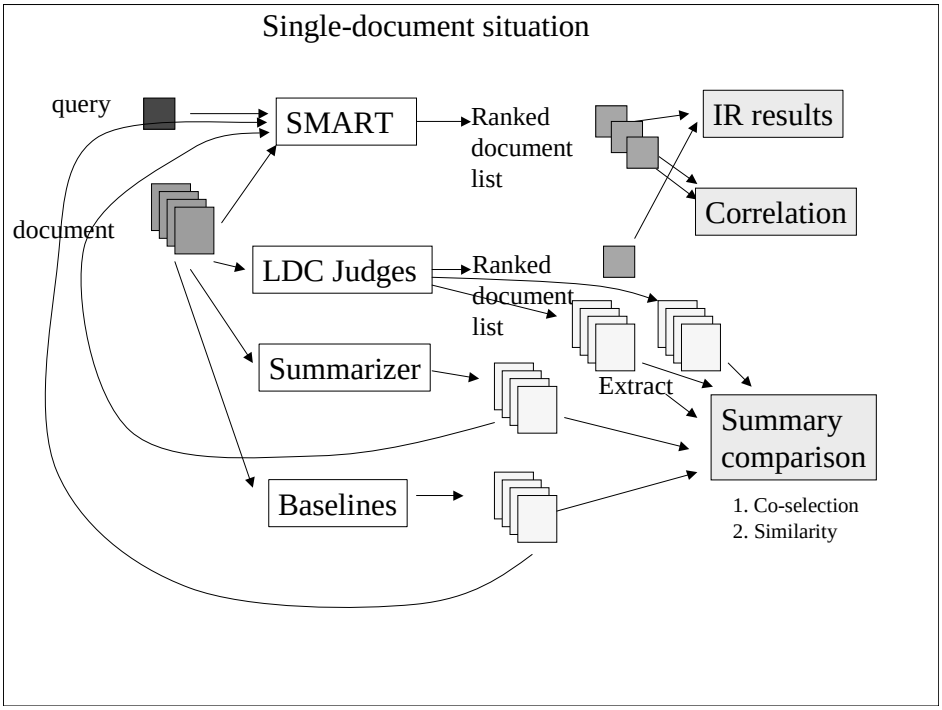


Figure 2.12: Single-document pipeline

The next set of experiments is multi-lingual. Our IR engine can run Chinese queries on the Chinese part of the corpus – provided we have translated the (originally English) queries into Chinese. The translation can be done automatically or manually. As the corpus contains sets of parallel documents with identical meaning, this setup allows to compare English IR performance to Chinese IR performance. Next, Chinese summaries can be created using the Chinese summarizers, and the same step as described below (indexing of Chinese summaries) can be performed.

We also use the 400 extracts provided by the LDC judges (40 queries X 10 documents each). For each document, three different humans create extracts for it independently. This means that we first have to compare

these summaries with each other, in order to establish in how far humans agree when they extract "most relevant" sentences. This is signified in on the right hand side of figure 2.12 as "summary comparison". As we will motivate later, there are several different ways how this summary comparison can be performed, for instance using co-selection or content-based similarity.

The summarizers also create extracts for the 400 documents, and these extracts are compared to the human extracts in the same way (by co-selection or content-based).

We used different summarizers (MEAD and WebSumm), and we also consider two different baseline systems, random extracted sentences and lead-based sentences. These are treated like summarisers and compared to the human judgements.

2.4.2 Multi-document case

Figure 2.13 shows the situation for multi-document summaries. For clusters of 10 documents each, for 40 queries, our judges create two types of summaries: extracts (by assigning sentence relevant grades to each sentence in the documents, effectively extracting sentences), and hand-written summaries of three different lengths (50, 100 and 200 words).

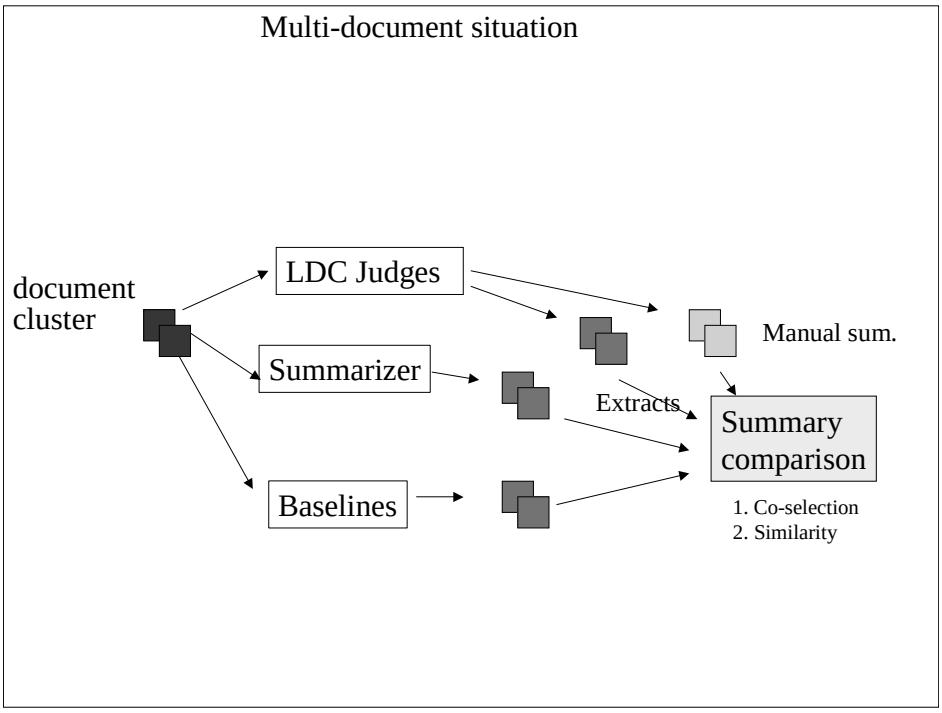


Figure 2.13: Multi-document pipeline

Our setup allows to compare human and machine-created multi-document extracts by co-selection comparison. Moreover, similarity-based comparison can be run on both extracts and summaries. However, in the multi-document scenario it makes no sense to compile relevance correlation, as there is no appropriate list of documents to compare the ranks to. In the single-document case, the ranks of full documents provided this appropriate comparison, but here, this is not possible.

Figure 2.14 presents seven different 10% multi-document extracts produced from the same cluster (Cluster 125). Note that order within a column is not relevant. One sentence with high salience is sentence 2 from article 19980306_007, which is chosen by several judges and several automatic systems. Later chapters will describe exactly how the comparisons are performed.

MEAD	LEAD	RANDOM	JUDGE1	JUDGE2	JUDGE3	ALLJUDGES
19980306_007:2	19980306_007:1	19980306_007:4	**19980306_007:2**	19980306_007:1	19980306_007:15	**19980306_007:2**
19980306_007:15	**19980306_007:2**	19980306_007:6	19980306_007:3	**19980306_007:2**	19980306_007:17	19980306_007:15
19980306_007:26	19980430_016:1	19980306_007:19	19980306_007:4	19980306_007:18	19980430_016:1	19980430_016:13
19980306_007:27	19980430_016:2	19980306_007:22	19980306_007:6	19990425_009:1	19980430_016:2	19980430_016:16
19980430_016:17	19990211_009:1	19980430_016:1	19980306_007:7	19990425_009:2	19980430_016:13	19990425_009:1
19980430_016:20	19990211_009:2	19980430_016:3	19980306_007:9	19990729_008:12	19980430_016:14	19990425_009:2
19980430_016:38	19990218_009:1	19980430_016:20	19980306_007:11	19990802_006:2	19980430_016:16	19990425_009:3
19990211_009:2	19990218_009:2	19980430_016:24	19980306_007:12	19990802_006:6	19980430_016:17	19990425_009:7
19990211_009:4	19990218_009:3	19980430_016:42	19980306_007:13	19990802_006:8	19980430_016:19	19990425_009:8
19990211_009:6	19990425_009:1	19990218_009:14	19990425_009:7	19990802_006:9	19990211_009:3	19990729_008:8
19990218_009:4	19990425_009:2	19990425_009:18	19990425_009:10	19990802_006:13	19990218_009:2	19990802_006:8
19990425_009:2	19990425_009:3	19990729_008:4	19990802_006:7	19990802_006:16	19990218_009:4	19990802_006:9
19990425_009:6	19990729_008:1	19990729_008:13	19990802_006:8	19990829_012:1	19990425_009:1	19990802_006:10
19990425_009:7	19990729_008:2	19990802_006:19	19990802_006:9	19990829_012:2	19990425_009:3	19990802_006:13
19990425_009:9	19990802_006:1	19990802_006:23	19990802_006:10	19990927_011:1	19990425_009:8	19990802_006:16
19990425_009:13	19990802_006:2	19990829_012:16	19990829_012:2	19990927_011:2	19990425_009:12	19990829_012:2
19990729_008:3	19990829_012:1	19990927_011:11	19990829_012:5	19990927_011:10	19990729_008:8	19990829_012:6
19990729_008:8	19990829_012:2	19990927_011:14	19990829_012:6	19990927_011:11	19990802_006:13	19990829_012:13
19990729_008:13	19990927_011:1	19990927_011:18	19990829_012:12	19990927_011:12	19990829_012:2	19990927_011:11
19990802_006:3	19990927_011:2	19990927_011:21	19990829_012:13	19990927_011:13	19990829_012:6	19990927_011:12
19990802_006:16	19990927_011:3	19990927_011:26	19990927_011:4	19990927_011:18	19990829_012:13	20000408_011:1
19990802_006:17	20000408_011:1	20000408_011:15	19990927_011:5	19990927_011:20	19990927_011:14	20000408_011:2
19990829_012:7	20000408_011:2	20000408_011:20	19990927_011:6	19990927_011:21	20000408_011:13	20000408_011:4
19990927_011:9	20000408_011:3	20000408_011:21	20000408_011:2	20000408_011:1	20000408_011:15	20000408_011:5

Figure 2.14: Seven 10% extracts produced from the same cluster

2.5 Manual summaries in the framework of the workshop

In this workshop, we do not produce automatic non-extractive summaries. Nevertheless, human summaries can be an extremely useful resource for comparison of manual extracts and automatic extracts in a content-based way. These evaluation measures give, in our opinion, a better understanding of the quality of extracts.

The LDC judges also wrote multi-document summaries for each cluster at 50, 100, and 200 words (independently of the size of the documents). As human summary writing by trained professionals is very expensive, we decided against single-document summaries. It would not have been possible to provide summaries of all 400 documents by several judges (and several compression rates). However, our judges found writing multi-document summaries to be a natural task. They followed our slight variation of the DUC guidelines (DUC2000, 2000), cf. next section), to do so. These texts provide a different gold standard for multi-document summaries in some of our experiments; we call them "human summaries". Human summaries are only available in English.

2.5.1 The Document Understanding Conference

The Document Understanding Conference (DUC) also employs human summaries as gold standards. Its first competitive conference (Spring/Summer 2001) was set up as a first information gathering process and pilot study of multi-document summarization.

Sixteen sites participated in the evaluation. The training data consists of 30 clusters of 8 - 16 documents each from the TREC collection, ie., newstories from different newspapers or news agencies such as the Wall Street Journal, the San Jose Mercury, and Associated Press. The clusters center around one "event" and its follow-up events, whereby the definition of an event differed between:

- One event happening at one time and place, e.g. the eruption of Mount Pinatubo.
- Several (more or less connected) events centered around one person, e.g. the career of Alan Greenspan
- Several unrelated events of the same type, e.g. reports of different fires on cruise ships or reports of sun eclipses.
- News items even more loosely related, e.g. news and book reports about the Antarctica, ranging from political decisions to expeditions to scientific projects on Antarctica.

These clusters were distributed with human-written summaries (one judge per task):

- One single-document summary per individual document, of 50 words length;

- Four multi-document summaries per cluster, of 400, 200, 100 and 50 words length.

After participants received the training data, the actual test was performed in July 2001, where another 30 clusters, previously unseen, were downloaded and summarized locally. Each participant then sent their summaries to NIST, where the evaluation was performed by comparison to the human-written summaries.

Evaluation was done by cutting the human-written summary (or model summary) into model units (MU), which could be a clause or a sentence, and by comparing these manually to the system summaries (or peer summaries). Peer summaries were automatically separated into peer units (PU). For comparison, the judge, who is the person as the one who wrote the summaries) used a tool called SEE, written by Chin-Yew Lin, which allows to display model and system summary in parallel and record the evaluation decision by the judge.

The judge decides for each MU if there are PUs covering it, either partially or totally. For non-covered PUs, the judges decide whether or not they should have been in the model summary. Final results are reported in precision and recall, which can be strict (full coverage only) or lenient (partial coverage too).

We designed our resource of manual summaries as close to the DUC data as possible, so that they can be useful to the community as additional material for the study of human and automatic summarization.

2.5.2 Guidelines for manual summaries in the workshop

The protocol we asked the summary writers to follow is as follows:

- Write a 200 word summary (+/- 5 words)
- Cut it down to a 100 word summary (+/- 5 words)
- Cut it further down to a 50 word summary (+/- 5 words)

The reason why we did not ask our judges to write 400 word summaries is that our texts are substantially shorter than the DUC texts and also of a different genre (administrative announcements of a municipal newspaper rather than proper news texts). Therefore, we suspected that it would be too difficult for the judges to produce summaries as long as 400 words. This was confirmed by the judges after having read the texts: they felt that there was not enough material to produce summaries as long as 400 words length.

In one respect our data is more informative than the DUC manual summaries: in the DUC setup there was *some* overlap between judges writing summaries (i.e. it is not the case that there was always only one summary per text), but the overlap was only in some part of the material. Our resource used three judges for each text, which provides more material for interannotator agreement – even if it is not guaranteed that each judge covered the entire data (there were more than three judges involved in our setup. Each of these judges wrote summaries for a subsection of the entire set).

We encouraged the judges, as the DUC guidelines had done, to reformulate sentences. Manual summaries were produced after the extracts and relevance judgements were created. Anecdotally, we heard from LDC that the judges found the task of writing summaries from scratch easier and more natural than the previous sentence extraction task.

Chapter 3

Information Retrieval of Documents and Summaries

Information retrieval (IR) aims at searching for useful or relevant documents in response to user queries. An IR model should be able to return a set of documents deemed relevant to the query and possibly rank the returned documents according to the estimated relevance.

Automatic summarization has been developed to alleviate the information overload problem. An interesting question is to see how useful summaries are for the information retrieval task. Specifically, automated summaries can be used for document surrogates for indexing. We wish to investigate and compare the IR performance using the automated summaries as well as entire documents. To this end, we introduce a new extrinsic measure for summarization called *relevance correlation* as discussed in subsequent chapters. The output of a retrieval run facilitates the computation of relevance correlation. Apart from mono-lingual retrieval, we also investigate cross-lingual retrieval of automated summaries due to the availability of a parallel corpus.

In this chapter, we will introduce a background on information retrieval models and a vector-space text retrieval engine SMART (Buckley, 1985) and XSMART. We will describe a cross-lingual retrieval technique employing phrasal translation and term disambiguation.

3.1 Information Retrieval Models

An information retrieval (IR) model is characterized by a quadruple $[D, Q, F, R(q_i, d_j)]$ (Baeza-Yates and Ribeiro-Neto, 1999) where

- D is a set composed of logical views (or representations) for the documents
- Q is a set composed of logical views (or representations) for the user information needs (queries).
- F is a framework for modeling document representations, queries, and their relationships.
- $R(q_i, d_j)$ is a ranking function which associates a real number with a query $q_i \in Q$ and a document representation $d_j \in D$.

Sample queries are shown in Figure 3.5. In an IR model, we need to design document and query representations. Different models have been proposed and some common ones are Boolean model, vector-based model, and probabilistic model. Typically each document is represented by a set of *index terms* characterizing the content of the document. Index terms can be words or phrases extracted or derived from the content.

The Boolean model is a simple and intuitive retrieval model. In this model, index terms are collected from the document and queries are specified as Boolean expressions. Despite this model has been used in many early retrieval systems due to its simplicity, there are several disadvantages. The first drawback is that the output of the retrieval result is a binary decision without any degree of relevance or partial match. The second drawback is that some information needs cannot easily be represented by Boolean queries.

To address the problem of binary relevance in Boolean models, vector-based and probabilistic models make use of real-valued weights for index terms. The result of a retrieval process is a ranked list of documents in decreasing order of relevance to the query.

IR models were developed originally in mono-lingual settings where the languages of the query and the documents are the same. For example, one can conduct retrieval on English documents using English queries. Likewise, one can conduct retrieval on Chinese documents using Chinese queries. In our project, the corpus contains both English and Chinese documents. Clearly, we need an IR engine that can handle both English and Chinese.

The original SMART could only process English documents. We made some enhancements to it so that it can index and retrieve both English and Chinese documents. More detailed description of SMART and our enhancements are given in the next section.

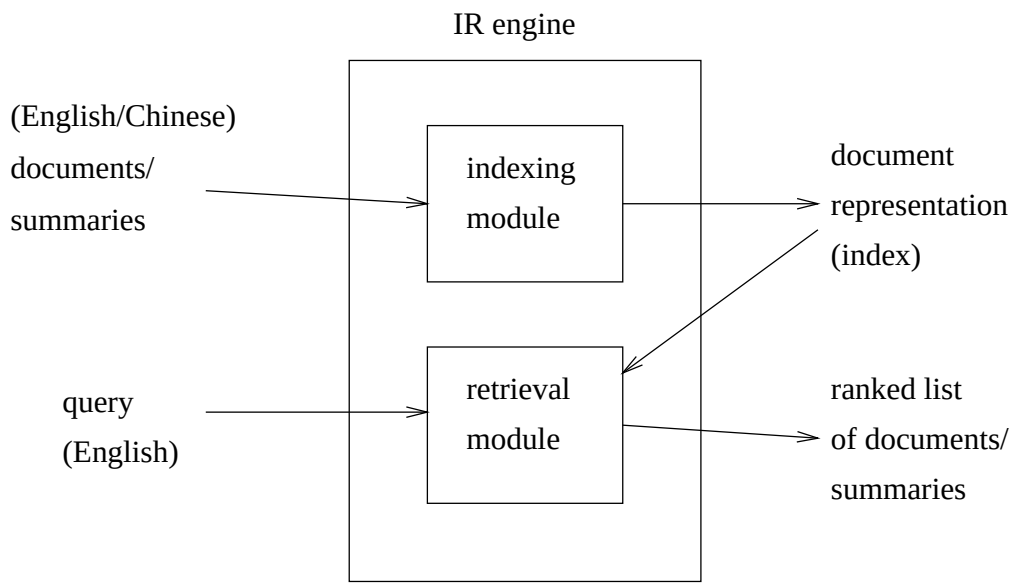


Figure 3.1: Information Retrieval Engine

Figure 3.1 illustrates the major tasks involved in our project. We made use of a collection of documents (or summaries). The first step was to index the collection. The output of this step was an index and related files. The output of the retrieval process was a ranked list of relevant documents (e.g., Figures 3.2).

Recently, there has been a surge of interest in developing cross-lingual retrieval techniques. In addition to mono-lingual retrieval, our project also investigated cross-lingual retrieval in which we retrieved Chinese documents based on English queries.

In addition to mono-lingual retrieval, we also investigated cross-lingual retrieval. Our approach first conducted query translation so that the translated query was expressed in the same language as the document/summary collection. Then we performed retrieval in a similar fashion as the mono-lingual one.

The query translation process mainly consists of two components. The first component is phrasal translation and the second component is term disambiguation. These two components will be described in more details in later sections.

3.2 SMART and Chinese SMART (XSMART)

SMART is a vector-based text retrieval engine originally developed by Salton and McGill (1983) and maintained by Buckley (1985). The framework of the vector-based model consists of vectors and operations on vectors. Precisely, each document D_i is represented as:

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{ik})$$

where each element d_{ij} denotes the weight of the index term T_j in the document D_i. Similarly, a query Q_m is represented as:

$$Q_m = (q_{m1}, q_{m2}, \ldots, q_{mk})$$

where each element q_{mj} denotes the weight of the index term T_j in the document Q_m.

There are a variety of methods for determining the weights of the elements on the vectors. A simple method is just to use the term frequency as the weight. Term frequency, f_j for the term T_j is defined as the number of occurrence of this term in a certain document. Another method is to consider the inverse document frequency (IDF) statistics in addition to term frequency. The basic form of inverse document frequency of a term T_j can be computed as:

$$log(\frac{N}{n_j})$$

where n_j is the document frequency of the term T_j in the document collection and N is the total number of documents in the collection. In SMART, users can specify different variants of weighting strategies based on term frequency and inverse document frequency. In our project, we made use of the following weighting scheme:

$$(0.5 + \frac{0.5 f_j}{\max_f}) log(\frac{N}{n_j})$$

where $\max_f$ is the maximum term frequency of a particular document.

The vectors were normalized for subsequent processing. A common normalization scheme known as cosine normalization was used. For example, suppose the weight after considering term frequency and inverse document frequency of a term in document D_i was w_{ij}. Then the final weight after cosine normalization was:

$$\frac{w_{ij}}{\sqrt{\sum_j w_{ij}^2}}$$

The weight q_{mj} in the query was calculated in a similar way.

During the retrieval process, a similarity score was computed between each document D_i and the query Q_m as follows:

$$\text{Score} = D_i \times Q_m$$

The output was a ranked list of documents sorted by decreasing order of similarity.

The original SMART supports all the above representation and model. However, it can only handle English documents. We had to make an enhancement to the way it reads a token so that it can deal with double-byte Chinese characters. We refer the enhanced version of SMART with bilingual features as XSMART

Another issue is the Chinese word segmentation problem. This problem arises due to the lack of word boundaries in Chinese texts. Our word segmentation program was derived from the program written by Peterson obtained from http://www.mandarintools.com. The original program only handles Chinese texts encoded in GB. We modified it so that it can handle texts encoded in BIG5.

A sample retrieval results for full-length documents is given below:

```
<?xml version='1.0'?>
<!DOCTYPE DOC-JUDGE SYSTEM "/export/ws01summ/dtd/docjudge.dtd">
<DOC-JUDGE QID="Q-241-E" SYSTEM="SMART" LANG="ENG">
  <D DID="D-20000126_008.e" RANK="1" SCORE="135.00" CORR-DOC="D-20000126_012.c"/>
  <D DID="D-19980625_007.e" RANK="2" SCORE="99.00" CORR-DOC="D-19980625_006.c"/>
  <D DID="D-19990126_017.e" RANK="3" SCORE="98.00" CORR-DOC="D-19990126_018.c"/>
  <D DID="D-19981007_018.e" RANK="4" SCORE="91.00" CORR-DOC="D-19981007_023.c"/>
  <D DID="D-19980121_004.e" RANK="5" SCORE="78.00" CORR-DOC="D-19980121_009.c"/>
  <D DID="D-19971016_004.e" RANK="6" SCORE="72.00" CORR-DOC="D-19971016_005.c"/>
```

Figure 3.2: Sample retrieval for full documents

A sample retrieval results for lead-based summary (5%) is given below:

```
<?xml version='1.0'?>
<!DOCTYPE DOC-JUDGE SYSTEM "/export/ws01summ/dtd/docjudge.dtd">
<DOC-JUDGE QID="Q-241-E" SYSTEM="SMART" LANG="ENG">
  <D DID="D-20000126_008.e" RANK="1" SCORE="14.00" CORR-DOC="D-20000126_012.c"/>
  <D DID="D-19991214_002.e" RANK="2" SCORE="11.00" CORR-DOC="D-19991214_001.c"/>
  <D DID="D-19980810_006.e" RANK="3" SCORE="10.00" CORR-DOC="D-19980810_003.c"/>
  <D DID="D-19990505_028.e" RANK="4" SCORE="9.00" CORR-DOC="D-19990505_014.c"/>
  <D DID="D-19980115_009.e" RANK="5" SCORE="9.00" CORR-DOC="D-19980115_013.c"/>
```

Figure 3.3: Sample retrieval for summaries

3.3 Phrasal translation

Phrasal translation is the first component in our query translation approach. Recall that the query is in English. The idea is to attempt to locate English phrases in the query and conduct translation on the phrases as far as possible. A basic resource we used in the phrasal translation process is a combined phrasal/word bilingual lexicon. After such a lexicon was constructed, it was used for conducting translation.

We made use of two resources to construct the combined phrasal/word bilingual lexicon. The first resource was a Chinese-English bilingual dictionary v 2.0 obtained from LDC. This lexicon is in Mandarin dialect encoded in GB and contains about 128,000 entries. Each entry in this lexicon contains a number of English translations for a particular Chinese word/term. The English translations are in words or phrases. There was another English-Chinese bilingual dictionary prepared by LDC. This dictionary contains Chinese translations for a particular English word. Since the Chinese-English bilingual dictionary contains English phrases, this dictionary is more suitable for our purpose of English query translation. Therefore, we decided to use this dictionary instead of the other one. We conducted some processing on this dictionary to produce the desired bilingual lexicon. Basically, we extracted all the English words and phrases from the dictionary and produced a new lexicon sorted by the extracted English terms. Each entry in the lexicon consists of an English word or phrases together with its Chinese translations. The second resource was a bilingual term lexicon derived from the Chinese-English Translation Assistance (CETA) dictionaries. We merged these two lexicons and produced the final English-Chinese lexicon consisting of about 210,000 entries.

Given an English query, we looked for phrases that match the bilingual lexicon as far as possible. The algorithm processes each text segment separated by punctuation one by one. It starts with the beginning of the segment. It attempts to match the phrases in the lexicon starting with the first word in the query. If one or more phrases are matched, it selects the longest phrase for translation. If no phrase is matched with the lexicon, then it looks up a list of translations for the single English word. Term disambiguation will be conducted to select one or a few good translations from the list. The next step is to repeat the same processing starting from the next available English word.

3.4 Term disambiguation

Given an English phrase or a word obtained by the phrasal detection algorithm described above, the query translation component looks up the corresponding entry in the combined bilingual lexicon. Normally, a number of Chinese translation terms exist in the lexicon. The number ranges from 1 to 70. The next problem is to select one or a few good translations. This is the main objective of term disambiguation task.

We tackle this problem by considering the neighboring context of the term as shown in Figure 3.2. We first introduce some notations. Let E_i be the English term to be considered and its set of Chinese translations in the lexicon be $\{C_{i,1}, \ldots, C_{i,k_i}\}$. Likewise, let E_{i+1} be the English term just after E_i in the query; the set of Chinese translations for E_{i+1} be $\{C_{i+1,1}, \ldots, C_{i+1,k_{i+1}}\}$. For each possible combination of the Chinese translation $(C_{i,x}, C_{i+1,y})$, we computed a score which relates to the co-occurrence of these two Chinese terms in a sentence. Specifically, the co-occurrence statistics is calculated as the number of sentences in which these two terms co-occur divided by the total number of sentences in the training corpus. We used the whole Chinese document collection of the Hong Kong News Corpus as the training corpus. The highest k pairs of translation terms are extracted and become the output of the translated query. After the translated query is obtained, it can be used for conducting retrieval of the target language which is Chinese in our experiments.

The output of a retrieval run facilitates the computation of relevance correlation which is a new measure for summarization quality. The retrieval performance can be measured by traditional recall and precision. Both the

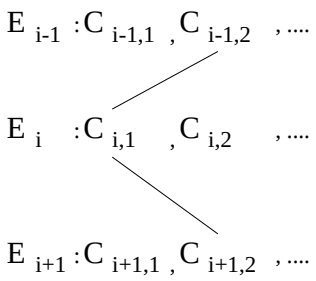


Figure 3.4: Term Disambiguation

```
Sample English Query

<?xml version='1.0'?>
<!DOCTYPE QUERY SYSTEM "dtd/query.dtd">
<QUERY QID="Q-241-E" QNO="241" TRANSLATED="NO">
<TITLE>
Fire safety, building management concerns
</TITLE>
</QUERY>


Sample Chinese Query

<?xml version='1.0'?>
<!DOCTYPE QUERY SYSTEM "dtd/query.dtd">
<QUERY QID="Q-241-C" QNO="241" TRANSLATED="NO">
<TITLE>
防火 意識 大廈 管理
</TITLE>
</QUERY>


CNPRINT WED 01-MAY-02 15:27:33 HKT                    1.
```

Figure 3.5: Sample Queries

relevance correlation and traditional retrieval performance measures will be discussed in subsequent chapters.

Chapter 4

Extractive Summarization

As a human activity, the production of summaries is directly associated with the processes of language understanding and production: a source text is read and understood to recognize its content which is then compiled in a concise text. In order to explain this process, several theories have been proposed and tested in text linguistics, cognitive science and artificial intelligence including macro structures (Kintsch and van Dijk, 1975; van Dijk, 1979), history grammars (Rumelhart, 1975), plot units (Lehnert, 1981) and concept/coherence relations (Alterman and Bookman, 1990). Several methods and theories have been applied to automatic text summarization research including the use of statistical measures, sentence position, cue and title words (Luhn, 1958; Edmundson, 1969; Kupiec et al., 1995; Brandow et al., 1995); partial understanding using conceptual structures (DeJong, 1982; Tait, 1982); bottom-up understanding, top-down parsing and automatic linguistic acquisition (Rau et al., 1989); recognition of thematic text structures (Hahn, 1990); cohesive properties of texts (Benbrahim and Ahmad, 1995; Barzilay and Elhadad, 1997) or rhetorical structure theory (Ono et al., 1994; Marcu, 1997). In this chapter, we briefly describe some approaches to text summarization research, and the summarization systems and algorithms used during this research.

4.1 Literature Review

We will now present a brief overview of prior work in text summarization both single and multi-document. While the review is incomplete we will try to cover most of the important work being carried out in the area.

4.1.1 Surface Level Approaches

Classical approaches to text summarization include the use of surface level indicators of information relevance and corpus statistics that can be applied to unrestricted text, here we review features usually employed.

Luhn (1958) developed the first sentence extraction algorithm which uses term frequencies to measure sentence relevance. The idea been that when writing about a given topic a writer will repeat certain words as the text is developed . So, term relevance is considered proportional to its in-document frequency. Luhn's algorithm filters terms using using a stop-list and computes term frequencies by aggregating terms together based on orthographic similarity. These term frequencies are later used to score and select sentences for the summary.

In general, when computing term relevance, function words like determinants and conjunctions, that are always highly frequent in any document are omitted. When dealing with a text collection in a certain field (e.g., Computer Science) it is likely that all documents share common terms in that field (e.g., computer, algorithm) while others are less frequent in the whole collection (e.g., distributed system). In these cases, the relevance of a term in the document is also inversely proportional to the number of documents in the collection containing the term (Salton, 1988). The normalized formula for term relevance is given by $tf_i * idf_i$, where tf_i is the frequency of term i in the document and idf_i is the inverted document frequency (that can be computed by $\log(N/dtf_i)$ where N is the number of documents in the collection and dtf_i is the number of documents containing term i). Relevant terms are those whose score are above a given threshold. Given term relevance, sentence scores can be

computed in a number of ways including the computation of a score based on the number of relevant terms the sentence contains, or the sum of scores of the relevant terms in the sentence or the number of clusters of relevant terms the sentence contains. As an alternative to measuring term relevance, concept relevance can be measured using WordNet (Lin and Hovy, 1997; Hovy and Lin, 1999) (an occurrence of the concept 'fruit' is counted when the word "fruit" is found as well as when an hypernymy of "fruit" is found). Term distribution has been shown less useful for sentence selection than other surface level features.

The relative position of a sentence in the document has been shown to be a good indicator of sentence relevance, Baxendale (1958) showed that first and last sentences of paragraphs are usually judged as highly content bearing: in a corpus of 200 paragraphs it was found that in 85% of the paragraphs the topical sentence occurred in first position and in 7% of the paragraphs the topical sentence occurred in last position. Recently, Lin and Hovy (1997) have developed algorithms for the automatic identification of position likely to yield good summary sentences: the Optimal Position Policy, a list that indicates in what ordinal positions in the text high-topic-bearing sentences occur. Not only the physical position of the sentence but also the 'logical' position of a sentence as a member of a particular 'conceptual' section is important for sentence extraction. Saggion (2000) has shown that in a corpus of abstracts written by professional abstractors more than 70% of the information for abstracts comes from introduction, conclusion, main title and section headings of the source document.

The presence of certain cue-words is also a surface level indicator of sentence relevance. One can consider that words like "important" and "relevant" point to *a priori* important information in sentences and so their presence can be used as a clue for sentence relevance. Here some cue-words are classified as "positive" (such as "important") while others are classified as "negative" (such as "believe"), the latter can be used for sentence deletion. In scientific texts, cue words or phrases like "in this paper", "in conclusion", "the results", etc. have been extensively used to locate sentences for abstracts (Edmundson, 1969; Paice, 1981; Teufel and Moens, 1999; Saggion, 1999), these kind of constructs generally called 'indicative phrases' signal information central to the scientific text. Sentences containing those indicators enriched with adjacent sentences have been used as indicative abstracts. While the task of constructing relevant lists of cue-words is time consuming and highly dependent on the domain, Teufel (1998) has shown how the identification of cue-phases can be accomplished in a semi automatic fashion.

Edmundson (1969) studied how combination of different linguistic and structural features affect the co-selection ratios between automatic abstracts and ideal abstracts. The work investigates the presence of pragmatic words (cue method), title and heading words (title method), and structural indicators (location method), as additional features for sentence worthiness. Edmundson demonstrates that a combination of cue, title, and location methods produces the highest mean co-selection score.

4.1.2 Trainable Summarization

Kupiec et al. (1995) implemented a Bayesian classifier that computes the probability that a sentence in a source document should be included in a summary. In order to train the classifier they used a corpus of of 188 pairs of full documents/summaries (written by professional abstractors) from scientific fields. The features (all discrete) used in order to represent sentences were: sentence length (true if sentence length greater that threshold), phrase structure (true if the sentence contains particular cue-words), paragraph (paragraph initial, paragraph final or paragraph medial), thematic word (sentences are scored according to the frequencies of their content words and high scored sentences receive a true value for this feature), uppercase word (same as before but only for non initial uppercase words). The probability that a sentence should be selected is:

$$P(s \in E|t_1, ..., t_k) = \frac{P(s \in E) * P(t_1, ..., t_k|s \in E)}{P(t_1, ..., t_k)} \tag{4.1}$$

where:

$P(s \in E \mid t_1, ..., t_k)$	probability that sentence s is contained in the extract given the features $t_1, ..., t_k$
$P(s \in E)$	probability that s is selected
$P(t_1, ..., t_k \mid s \in E)$	probability of the set of features $t_1, ..., t_k$ in the extract
$P(t_1, ..., t_k)$	probability of the set of features $t_1, ..., t_k$ in the text

The estimation of parameters (assuming independence) is as follows:

$$P(t_1, ..., t_k \mid s \in E) = \prod_{i=1}^{k} P(t_i \mid s \in E)$$
$$P(t_1, ..., t_k) = \prod_{i=1}^{k} P(t_i)$$

Using this probabilistic approach they found that location-based features gives the best performance and a combination of location, cue-word, and sentence length give the highest co-selection mean.

4.1.3 Cohesion-based summarization

The main drawback of extractive methods is that they usually fail to capture the relations between the concepts in texts. Anaphoric expressions (pronouns and definite noun phrases), that are used to refer back to events and entities in the text need their antecedents in order to be understood. When sentences containing anaphoric links are extracted without the previous context the resulting summary become unintelligible. Text cohesion (Hallyday, M.A.K. and Hasan, Ruqaia, 1996) involves relations between words, word senses, or referring expressions, which determine how tightly connected a text is. Cohesive properties of the text have been explored by different approaches to text summarization in order to cope with the above problems.

Barzilay and Elhadad (1997) compute lexical chains as the basis for text summarization. This approach with be described in Section 4.3.4. Mani and Bloedorn (2000) also explore the use of cohesion relations between proper names, to construct user-focused summaries for multiple articles. Their approach will be described in Section 4.3.2

Tele-pattan (Benbrahim and Ahmad, 1995) is a text summarization system based on the notion of lexical cohesion. An analysis of lexical cohesion, by counting repetitions, synonyms, super-ordinate terms and paraphrases, leads to the establishment of a network of sentences, some tightly bonded to each other, while others have weak bonds or not bonds at all. The density of bonds in the network and the distribution of the bonds is used to decide which sentence of the text are theme opening, or closing or marginal. Summaries can be generated that open, continue and close a given topic.

4.1.4 Rhetoric-based summarization

Rhetorical Structure Theory is a descriptive theory about text organization. The theory consists of a number of rhetorical relations that tie together text spans, and a number of recursive schemas specifying how texts are structurally composed in a tree-like representation. Most relations are binary and asymmetric: they tie together a nucleus (central to the writer's goal) and a satellite (less central material). Ono et al. (1994) and Marcu (1997) made use of RST as the basis for text summarization. The approach consist on the construction of a rhetorical tree based on the presence of explicit discourse markers and the use of heuristic rules to decide for the best rhetorical tree for a given text. In the case of Marcu the minimal unit of analysis is the clause while in Ono et al's is the sentence. After the tree is obtained text units have to be extracted for the summary. In Ono et al's approach sentences are penalized according to their rhetorical role in the tree. A weight of 1 is given to satellite units and a weight of 0 is given to nuclei units. The final score of a sentence is given by the sum of weight from the root of the tree to the sentence. Sentences can be ordered in ascending order of scores and used as the basis for constructing the summaries. In Marcu's approach, each parent node identifies its nuclear children as salient, promoting their children to their level. The process is recursive down the tree. The salience score of a clause is given by the level it obtained after promotion. The scores are used to produce a ranked list of clauses that can be used as the basis

for summarization.

In the context of the scientific article, Rino and Scott (1994) have addressed the problem of coherent selection for text summarization using some aspects of RST, but they depend on the availability of a complex meaning representation which in practice is difficult to obtain from the raw text. Relevant work in rhetorical classification for scientific articles, which is the first step towards the production of scientific abstracts, is due to Teufel and Moens (1997), who used statistical approaches borrowed from Kupiec et al. (1995). Teufel and Moens have developed a program able to instantiate a rhetorical scheme with the following components: background, topic, related work, purpose, solution, result and conclusion. The instantiate scheme contains enough rhetorical information to determine the rhetorical contribution of all and only the abstract-worthy sentences in the text. In addition to basic features such as location, word distribution, etc., they compiled a list of cue phrases that were assembled into five classes based on occurrence frequencies in ideal summaries. The authors have shown that superficial features of the text can be effectively used in rhetorical classification.

4.1.5 Knowledge Intensive Approaches

Knowledge intensive approaches are based on the extensive encoding of world knowledge about specific situations. These methods base the selection of information not on the surface level properties of the text, but on expected information about a well known situation. They are characterized for including text generation techniques that allow the production of compact, cohesive and coherent texts.

FRUMP (DeJong, 1982) uses sketchy-scripts based on scripts (Shank and Abelson, 1977), rich knowledge representation formalisms used to model stereotypical situations in a domain (e.g., 'earthquake', 'kidnapping'). Sketchy-scripts contain only the 'key' information to be expected in a situation. Script activation is based on a number of indexing word senses associated to the script, inferences are made on the basis of expectations and subsequent matching against the input text. Instantiated slots are used to generate summaries in several languages. The main problem with this approach is the difficulty to adapt scripts to completely new domains and the fact that the scripts are not able to deal with "unexpected" information.

4.1.6 Information Extraction and Summarization

Information Extraction is the process of mapping natural language into predefined, structured representations, that when instantiated represent the key information from the original source (Gaizauskas et al., 1997). Concept-based abstracting (CBA) (Jones and Paice, 1992; Paice and Jones, 1993) is an Information Extraction approach to text summarization. CBA is used to produce abstracts of technical articles in specific domains, for example, in the domain of agriculture. Semantic roles such as species, cultivar, high level property, low level property, etc. are first identified by the manual analysis of a corpus, and then patterns are specified that account for stylistic regularities of expression of the semantic roles in texts. These patterns are used in an information extraction process that instantiates the semantic roles. CBA uses a fixed canned template for generation. The method was mainly used to produce indicative abstracts, though some informative content is included in the form of extracted sentences containing results and conclusions (Paice and Oakes, 1999).

4.1.7 Summarization by Generation

While sentence extraction is a currently wide-spread, useful technique, more research in summarization now is moving towards summarization by generation. Jing and McKeown (2000) and Jing (2000) propose a *cut-and-paste* strategy as a computational process of automatic abstracting and a sentence reduction strategy in order to produce concise sentences. They have identified six "editing" operations in human abstracting: (i) sentence reduction; (ii) sentence combination; (iii) syntactic transformation; (iv) lexical paraphrasing; (v) generalization and specification; and (vi) reordering. Their algorithm for sentence reduction takes into account different sources of information to decide whether or not to remove a component from a sentence. The decision is taken based on: (i) the relation of the component to its context; (ii) the probability of deleting such a component (estimated

from a corpus of reduced sentences); and (iii) linguistic knowledge about the essentiality of the component in the syntactic structure. Saggion and Lapalme (2000a) produced 100 tables containing professional abstracts aligned (on the sentence level) with source documents. These alignments were used to identify on one hand, concepts, relations and types of information usually conveyed in abstracts; and on the other hand, valid transformations in the source in order to produce a compact and coherent text. The transformations include: (i) verb transformation; (ii) concept deletion; (iii) concept reformulation; (iv) structural deletion; (v) parenthetical deletion; (vi) clause deletion; (vii) acronym expansion; (viii) abbreviation; (ix) merge; and (x) split. In their corpus of alignments, 89% of the sentences from the professional abstracts included at least one transformation. Based on their corpus study they have developed a text summarization system that produces indicative-informative abstracts for technical articles. Their approach to text summarization is based on a superficial analysis of the source document and on the implementation of some text re-generation techniques. The analysis of the text consists on the instantiation of a number of indicative and informative templates by a pattern matching process. Their generation algorithm uses re-generation schemas based on the instantiated templates.

4.1.8 Multi-document Summarization

Automatic multi-document summarization (MDS) refers to the problem of producing an abbreviated version a set of "related" documents. The term "related" can mean different things in different situations. For example, documents can be related because they describe the same event (e.g., "the World Trade Center terrorist attack") and there is a need to produce a single piece of text that will reduce redundancy and will bring the information unique to each source. Documents can also be related, for example, because they are research articles referring to different proposed solutions to the same problem ("statistic methods for multi-document summarization" vs "symbolic methods for multi-document summarization"), and in that case, a multi-document summary should contain information about what solutions each text propose, and their differences in, for example, effectiveness of the antagonist solutions. Nowadays, the availability of on-line documents referring to the same information makes multi-document summarization a problem worth to solve. A MDS system should be able to cope with the problems of redundancy across documents as well as with the problem of identifying new and relevant information across documents. In the context of non-extractive MDS, the system should be able to combine information from different sources in a very compact text. Some characteristics unique to MDS: (i) higher compression rate than for single document summarization; (ii) need to deal with redundancy[1]; (iii) need to fuse the information, (iv) need to re-express the information probably using natural language generation techniques.

We follow Mani (2001) in the classification of approached to MDS: (i) morphological approaches are based on measures of vocabulary overlap to identify similarities across sources; (ii) syntactic methods use sentence structure to identify syntactic paraphrase; and (iii) semantic methods rely on a conceptual representation of the text, where differences in expression are mapped into the same meaning.

Morphological Approaches

Robust approaches come mainly from information retrieval, cosine similarity is a well known measure to compute how similar document representations are. Cosine similarity can be computed using the following formula (Salton, 1988):

$$cos(X,Y) = \frac{\sum x_i * y_i}{\sqrt{\sum (x_i)^2} * \sqrt{\sum (y_i)^2}}$$

Where X and Y are text representations based on a vector space. Salton et al. (1994) have used cosine similarity to identify passages related to unique topics but for single documents. Methods based only on vocabulary overlap fall off when related texts use different vocabulary (synonyms) and when systems do not solve coreference. Latent Semantic Analysis (Deerwester et al., 1990) is a technique that reduce the vector space in order to map "related" terms (or co-occurring terms in a collection) into the same dimension. This method was used for multi-document summarization (Ando et al., 2000). We explore the use of similarity measures for the purpose of evaluation (see next chapter).

[1] In long, scientific articles redundancy is used to make clear what the article is about: "In this paper we study X" when opening the research article and "In this paper we have studied X" when closing.

Radev et al. (2000) developed an earlier version of MEAD which, given an event cluster of related documents, from a topic detection and tracking system, produces summaries in the form of sentence extracts. MEAD use the cluster centroid, a set of words that are central to all the articles in the cluster, as a key feature for sentence selection. MEAD decides which sentences include in the extract based on the following features: the similarity of the sentence with the centroid, the similarity of the sentence with the first sentence in the document, and the position of the sentence in the document, where the similarity is computed using the cosine between two text elements.

Maximal Marginal Relevance (Carbonell and Goldstein, 1998) is a method to measure "relevant" novelty. Suppose a typical text retrieval engine that ranks document according to their relevance to a user-query. It is probable that documents in the top of the list are quite relevant to the query and well related among them, while documents down in the list, while still relevant to the query, are quite different from the documents on top. MMR uses feedback from the user (in the form of documents already examined) to re-rank and possible bringing to the top of the list documents that maximizing the dissimilarity with the previously seen are still relevant to the query. The formula used to compute "relevant novelty" is the following:

$$MMR(Q, R, S) = argmax_{D_i \in R \setminus S}\{\lambda(sim_1(D_i, Q)) - (1 - \lambda)max_{D_j \in R}(D_i, D_j)\}$$

where Q is the query, R is the set of retrieved documents, S is the set of documents scanned by the user, and $R \setminus S$ is the set difference between R and S. λ is a parameter that, when equals 1, gives preference to documents that are maximally relevant to the query and, when 0, gives preference to documents gives preference to documents that diverge from the already scanned by the user. When used in the context of MDS (Goldstein et al., 2000), text is segmented into passages (e.g., sentences, paragraphs, chunks, etc.), and the following features are used in the formula the similarity between the passage and the query, the coverage of the passage (how important the passage is), linguistic features of the passage (such as presence of named entities and position), and temporal information (document level). In order to reduce redundancy the features used are: cosine similarity with already selected passages, passages that come from cluster already present in the summary, and documents with passages already in the summary. The question of how to estimate the parameter λ, however remains.

Mani and Bloedorn (1999) explore cohesion relationships to construct user-focused summaries. They rely on a graph representation of the text where the nodes are terms and the edges are cohesion relationships such as repetition, synonymy, hypernymy, and coreference. Coreference is only limited to the case of proper names. A spreading activation mechanism uses the query terms to activate regions of the graph by first matching the query terms to the nodes, and then propagating weights through paths in the graph, in this way other terms are scored based on their relation with query-related terms. The activated regions of the graphs are compared for common and unique terms. Sentences are extracted based on an scoring mechanism that uses the term weights. Depending on the need to avoid redundancy unique or common terms can be favored.

Syntactic Approaches

Sentence structure can help in identifying syntactic paraphrase and thus similar information across documents. McKeown et al. (1999) describe a MDS system that identifies groups of paragraphs which all convey approximately the same information. They use a number of linguistic features to measure similarity including: word co-occurrence; noun phrase matching; synonymy matching; and verb semantic marching. These features are used to training a machine learning algorithm that identifies whether two paragraphs are similar. Based on a parser that produces a 'dependency-tree' encoding grammatical relations between sentence components, on each theme they compare dependency trees using a tree matching algorithm in order to identify intersections (same main verb and the same arguments). The authors did not mention co-reference resolution that is essential to identify identity between arguments. Intersections are used to feed a generation component where the input already contains the lexical items to be used, nevertheless when sentences come from different sources the choice of appropriate vocabulary can be a real challenge. While fluency is a matter of text, the authors report good performance in fluency when evaluating only isolated sentences.

Semantic Approaches

Semantically motivated approaches use some kind of meaning representation of the text. In the case of template based multi-document summarization, key pieces of information are represented through templates, automatically filled by an information extraction system tuned to a specific domain. The problem of producing a summary is re-stated as the problem of generating a single text from a set of templates that contain information about the same event. Radev and McKeown (1998) have developed SUMMONS, a knowledge-based multi-document summarizer that operates on a set of templates containing the key information extracted from texts on terrorism. The novelty of their approach consists on the development of a number of summary operators for content planning. These operators are used to combine information from two different templates and account for a variety of situations that may occur when the same event is described by different sources. The operators include among others: change of perspective, contradiction, agreement, etc. For sentence realization, the system uses a set of summarization phrases collected from corpora.

4.1.9 Research on non-extractive summarization

In this workshop, we create summaries by sentence extraction. But there are non-extractive summarization methods, ie. those that create a summary from sentence fragments, rather than extracting sentences. Instead of reproducing full sentences verbatim from the text, these methods either compress the sentences (Grefenstette, 1998c; Jing and McKeown, 2000; Knight and Marcu, 2000), or re-generate new sentences from scratch (Barzilay et al., 1999; McKeown et al., 1999). It is plausible that summaries produced this way resemble the human summarization process more than extraction does; however, if large quantities of text need to be summarized, sentence extraction is a more efficient method, and it is robust towards all kinds of input, even slightly ungrammatical one. Some of the non-extractive methods rely on successful parses as a precondition to producing summaries. Some approaches require extensive linguistic resources and others need to solve complex problems like coreference resolution.

4.2 The MEAD summarizer

MEAD is a centroid-based multi-document summarizer that was first introduced in Radev et al. (2000). The version of MEAD that we built from scratch during the workshop is modularized as three separate components: A feature extractor calculates a value for user-defined features of each sentence, a sentence scorer gives sentences values according to a linear combination of their features, and a sentence re-ranker changes the scores of sentences according to cross-sentence relationships. The summarizer and all of the features we used in the workshop operate in both English and BIG5 encoded Chinese.

4.2.1 Architecture of MEAD

Feature Extractor

MEAD's modularized feature extraction allows features to be calculated and stored separately from the summarization process. This offers two advantages. Users can write feature calculation scripts without having to worry about the rest of the MEAD architecture, and computationally expensive features need only be computed once. They can then be stored and reused as many times as needed. The features we used to perform the experiments described in this paper are as follows:

- Position: If a sentence is the nth sentence in the document, position $= \sqrt{\frac{1}{n}}$.
- Centroid: A centroid vector is a set of terms together with their frequencies in a document cluster. A centroid vector is calculated for a document cluster as follows:
  1. All of the terms with $tf * idf$ value greater than 3 are added to the centroid.
  2. If the total number of terms in the centroid is still less than 10, then the terms with the highest $tf * idf$ values are added until the centroid has 10 terms.

For a sentence consisting of terms $t_1, t_2, \ldots, t_n$, let $cen(t_i)$ be the value for t_i stored in the centroid vector for the cluster. Then if "Centroid" is the value of the Centroid feature, Centroid $= cen(t_1) * idf(t_1) + cen(t_2) * idf(t_2) + ... + cen(t_n) * idf(t_n)$. Because the Centroid feature for sentences can vary widely across document clusters, we normalize the Centroid feature value by dividing by the value of the highest-valued sentence in the cluster.

- Cosine (weighted tf*idf) with Query (used only in Query-based summarization): Where t_i indicates a term in both the sentence and the query, t_j indicates a term in the query, and t_k indicates a term in the sentence, the Cosine similarity with the query is

$$\frac{\sum_i ((IDF_i * Q_i) * (IDF_i * S_i))}{\sqrt{(\sum_j (IDF_j * Q_j)^2) * (\sum_k (IDF_k * S_k)^2)}}$$

- Length: Length is the number of terms in the sentence.

Sentence Scorer

With the exception of length, MEAD scores sentences using a linear combination of their feature values:

$$score(s) = w_1 * f_1(s) + w_2 * f_2(s) + \ldots + w_n * f_n(s)$$

The Length feature is used to give sentences a score of 0 if they are too short. For the experiments of the workshop, a sentence received a score of 0 if its length was less than 9. Since the other two features used in the workshop had values of less than one, the weights in the linear combination decided their importance relative to one another. The formula we used for generic summaries in the workshop then became:

$$score(s) = \left\{ \begin{array}{ll} Centroid(s) + Position(s) & length > 9 \\ 0 & length < 9 \end{array} \right\}$$

Different variations of this which included the query-based similarity feature were used for query-based summaries.

Sentence Re-ranker

After ordering sentences according to their scores, the MEAD re-ranker adds sentences to the summary iteratively, beginning with the highest-scoring sentence. At each iteration, it calculates the cosine similarity of the sentence to be added with each of the sentences already in the summary. If the similarity is higher than a given threshold, it discards the sentence and moves on to the next one, continuing until it has added the number of sentences corresponding to the summary length.

```
For each sentence s_i until # of sentences in
summary = max number needed for given summary length,
    For each sentence in summary, s_j,
        if s_i too similar to s_j,
            discard s_i
        else,
            add s_i
    End
End
```

Figure 4.1: The MEAD reranking procedure

4.2.2 The Centroid Feature

A MEAD cluster Centroid is a bag-of-words vector representation of the important concepts in a cluster. Given a cluster, MEAD decides for each term in the cluster whether or not to include it based upon its $tf * idf$ score. MEAD uses a 2-step iterative algorithm, adding the terms for each document to the cluster at the beginning of each iteration, and deleting the low-scoring terms for each document from the cluster at the end. This process is illustrated in Figure 4.2.

```
For each document d,
  For each term t in d,
    if t_i has not occurred in the centroid,
        add t_i to the centroid
    else,
        increment count(t_i) in the centroid
  End
  For each term t_j in the centroid
    if tf*idf(t_j) is below threshold,
       throw out t_j
    else,
       keep t_j
  End
End
```

Figure 4.2: Centroid Computation in MEAD

4.3 Other summarization methods used in the workshop

4.3.1 Baselines

Two baselines summarizers are used in order to produce a n% extract:

- Lead-based summarizer: n% sentences are picked up from the beginning of the text;
- Random summarizer: n% sentences are picked up at random[2]

4.3.2 Websumm

Websumm (Mani and Bloedorn, 2000) is a single and multi-document summarizer developed at MITRE. It can be used to produce generic and query-based summaries. Websumm uses a graph-connectivity model. The basic idea of representing texts in terms of graphs is that the topology of the graph will reveal something interesting about the salience of information in the text. In particular, a common Graph Connectivity Assumption is that nodes which are connected to lots of other nodes are likely to carry salient information. In Websumm, the nodes are occurrences of words or phrases, and the links are relations of repetition, synonymy, and coreference. Given such a graph representation for a document, the summarization algorithm takes a topic (a user's query) and produces topic-focused extracts by finding occurrences of query terms in the graph. A spreading activation algorithm then explores nodes related to query nodes in the graph. As the activation spreads, different term positions in the graph get different weights, creating a salience contour for the text. Sentences are then extracted up to the compression based on the weight of terms in them. In the case of a generic summary, the spreading activation search isn't carried out; instead, the system functions as a term frequency based summarizer, by using the weights in the original graph.

[2]The perl package Random is used.

4.3.3 Summarist

Summarist (Hovy and Lin, 1999) is a text summarization system developed at the Information Science Institute of the University of Southern California. Its goal is to produce summaries of multilingual input texts in both generic and query-based forms. It can process English and Chinese among other languages (Lin, 1999). The system is structured in three main modules: topic identification, topic interpretation, and generation. Before topic identification, the system identifies words, applies part-of-speech tagging and a morphological analysis, identifies multi-words phrases and computes term frequency and $tf * idf$ weights.

The topic identification module filters the input text to retain the most important topics. It uses a number of modules to associate an score to each sentences including a Optimal Position Policy (OPP) module to identify sentence positions most likely to yield good summary sentences (Lin and Hovy, 1997), a Cue Phrase module to identify sentences containing cue words, and a high-frequency indicator phrases module. Topic interpretation and generation have not been completed.

4.3.4 Lexical Chains

Lexical chains are a means of representing lexical cohesion among sequences of words. An algorithm for text summarization using lexical chains as text representation was introduced by Barzilay and Elhadad (1997). They used the WordNet database (Fellbaum, 1998) for determining cohesive relations (i.e., repetition, synonymy, antonymy, hyperonymy, and holonymy) between nouns and noun compounds identified by the processes of POS tagging and shallow parsing. Lexical chains are constructed on segments identified by a segmentation process and chains are merged across segments whenever they contain the same term with the same sense. Scores for lexical chains are determined on the basis of the number and type of relations in the chain. The strongest lexical chains are selected in order to construct the summary. Sentences are selected from the text based on a number of heuristics such as identifying sentences where the strongest chains are highly concentrated. In our research, we used the more recent implementation of the lexical implemented by Silber and McCoy (2000).

etrainees
Some sample summaries are shown in Figure 4.3.

SUMMARIST

The Director of Home Affairs, Mrs Shelley L, today (Friday) visited a group of new arrival r from the mainland who were taking a Job Search Skills Course to share their experience in joining the retraining programme and in seeking jobs in Hong Kong.

LEAD

New arrival retrainees share experience with DHA

The Director of Home Affairs, Mrs Shelley Lau, today (Friday) visited a group of new arrival retrainees from the mainland who were taking a Job Search Skills Course to share their experience in joining the retraining programme and in seeking jobs in Hong Kong.

MEAD

The Director of Home Affairs, Mrs Shelley Lau, today (Friday) visited a group of new arrival retrainees from the mainland who were taking a Job Search Skills Course to share their experience in joining the retraining programme and in seeking jobs in Hong Kong. Speaking after her visit to the Hong Kong College of Technology's Retraining Resource Centre, Mrs Lau said she was most delighted to see that new arrivals are now eligible to apply for any retraining course or programme funded by the Employees Retraining Board.

RANDOM

New arrival retrainees share experience with DHA

She was briefed by the Executive Director of the Employees Retraining Board, Mr Chow Tung-shan, and the Principal (Designate) of the Hong Kong College of Technology, Mr Chan Cheuk-hay, on the range of services provided at the retraining centre.

WEBSUMM

Speaking after her visit to the Hong Kong College of Technology's Retraining Resource Centre, Mrs Lau said she was most delighted to see that new arrivals are now eligible to apply for any retraining course or programme funded by the Employees Retraining Board. "Thus, the Government has extended the Employees Retraining Scheme to cover new arrivals from January 31 this year.

Figure 4.3: Sample summaries

Chapter 5

Evaluation Methods

The evaluation of text summarization systems is an emergent and difficult research topic, but the distinction between intrinsic and extrinsic methods as defined by Spark Jones and Galliers (1995) has been generally accepted. Whereas extrinsic evaluations measures how helpful summaries are in the completion of a given task, intrinsic evaluation measures the quality of the summary itself, e.g. by comparing the summary with the source document, by measuring how many "main" ideas of the source document are covered by the abstract or by comparing the content of the automatic summary with an ideal abstract (gold standard) produced by a human (Cole, 1995). Subtypes of intrinsic evaluation are: content evaluation (which assesses if automatic systems are able to identify the intended "topics" of the source document), and text quality evaluation (which assesses the readability, grammar and coherence of automatic summaries).

In actual research, the following practical evaluation solutions are dominant:

1. Comparison to ideal summary (intrinsic)
2. Subjective evaluation on a scale (intrinsic), and
3. Task-based evaluation (extrinsic)

For sentence extracts, comparison to an ideal extract is often performed by measuring co-selection: what proportion of sentences are in both the extract and the ideal sentences. The main evaluation metrics used in intrinsic evaluations are precision, recall and F-score (Firmin and Chrzanowski, 1999), which have been extensively used in the past to evaluate text summarization systems (Edmundson, 1969; Brandow et al., 1995; Marcu, 1997; Barzilay and Elhadad, 1997). Precision (P) is the number of correct answers given by the system divided by the number of answers the system produces. Recall (R) is the number of correct answers given by the system divided by the number of correct answers. F-score is a composite measure that combines precision and recall in the following formula:

$$\frac{(\beta^2+1)PR}{\beta^2 P+R},$$

where β is a weighting factor that favors precision when $\beta > 1$ and favors recall when $\beta < 1$.

In this work, we compare percent agreement of co-selection (also called accuracy), precision and recall of co-selected sentences, relative utility, and Kappa between selection decisions as co-selection measures.

Co-selection evaluation methods have some restrictions. For instance, they only work for extractive summarizers, as they compare entire sentences and only count as a match the exact same sentences. This ignores the fact that two summaries can contain the same information, but contain different sentences which express this information differently. Content-based metrics apply different notions of similarity to compare the actual words in a sentence, rather than the entire sentence as a string.

One advantage of similarity metrics is that they can compare both human and automatic extracts with human *abstracts*, i.e. coherent, newly written summaries of the documents rather than sentence extracts. To our knowledge, no systematic experiments about agreement on the task of summary writing have been performed before. In our experiments, the similarity measures are applied to 3 human summaries per topic. They answer the question of how much humans agree when summarizing. The same similarity measures are applied to 3 human extracts

per topic, which answers the question of how much they agree when extracting. We also apply the measures *between* human extracts and summaries, which answers the question if human extracts are more similar to automatic extracts or to human summaries.

Relevance correlation is our contribution as a new measure of summary quality. It shows the relative performance of a summary: how much does the performance of information retrieval decrease in comparison to the full texts.

Task-based evaluations (Mani et al., 1999a; Tombros et al., 1998; DUC2000, 2000) measure human performance using the summaries for a certain task (*after* the summaries are created). Although they can be a very effective way of measuring summary quality, task-based evaluations are expensive exercises requiring human subjects *during* the actual evaluation. As this was not possible due to the setup of the workshop, we do not consider task-based evaluation here. However, due to some similarities with our work, we discuss it here.

While our new measure relevance correlation is an extrinsic measure (as task-based evaluation is), there are differences between the two in other respects. Let us first describe a typical IR-based task-based summary evaluation scenario.

In a typical task-based evaluation such as in Tombros et al. (1998) or Mani et al. (1999a), humans have the task to decide how relevant a document is to a certain query. Their performance is measured and compared in two cases: one when they are shown summaries, the other, when they are shown the full document (ceiling condition) or a baseline (e.g. random or first sentences). Their performance is quantified in precision and recall and in reading time.

In Tombros et al. (1998) humans were asked to identify as many relevant documents as possible in a fixed time frame, on 50 randomly chosen TREC queries. Their results showed that query-based extracts allow users to see more documents (23 instead of 20) with higher precision (55% vs. 44%) and recall (66% vs. 50%) than baseline does; additionally, humans ask for full documents in less cases (1% vs. 24%).

SUMMAC (Mani et al., 1999a) is a large-scale, TIPSTER sponsored comparative task-based evaluation using 16 participating systems. Apart from a categorization and an experimental question-and-answer task, the main focus was on the IR task (called ad hoc task) with 20 TREC-queries.

Each site submits a fixed length summary (S1) and a variable length summary (S2) for each text, which are compared against baseline summaries (B) and full-text (F). 1000 documents are judged per subject (20 F, 20 B, 480 S1, 480 S2), and the measured variables are time and F-score. The experiment showed that summaries of 17% length result in same F-score accuracy as full texts, in about half the decision time (33 sec. per decision vs. 59 sec.) The F-score of the full texts was .67, of the baseline .42. However, the results could not distinguish between participant technologies. Human agreement is reported as: pairwise 69%, three way 54%, unanimous (14 judges) 17%. Kappa was .38.

Extrinsic methods for the evaluation of summaries are based on the premise that summaries are produced with a certain task in mind. That task could be question answering, classification, etc. Relevance correlation is another such metric.

The reason why we did not use task-based evaluation is that we do not have judges available during the workshop, i.e. we cannot elicit direct judgments on the summaries as the summaries are created *after* whichever judgments we need are created. Therefore, we cannot, as Tombros et al. and Mani et al. did, show them the summaries created by the summarization systems and ask them to perform a task on the basis of the information contained in the summaries. Instead, we have to use some relevance judgments which were created independently of the summaries and therefore, we chose the evaluation setup described in section 2.

5.1 Sentence co-selection

We compiled utility judgments into lists of binary decisions per sentence, asking whether or not a given sentence is included in a summary at a certain target length. We use the rates 5%, 10%, 20%, 30%, 40% for most experiments. For some experiments, we also consider summaries of 50%, 60%, 70%, 80% and 90% of the original length of the documents (Figure 2.11).

5.1.1 Percent agreement

Percent agreement (Figure 5.1) measures how many of the judges' decisions are shared between two judges.

Percentage agreement between two judges is defined as follows:

		Judge J1		
		Sentence in Extract	Sentence not in Extract	
Judge J2	Sentence in Extract	A	B	$A+B$
	Sentence not in Extract	C	D	$C+D$
		$A+C$	$B+D$	$N = A + B + C + D$

Figure 5.1: Contingency table on binary decisions

$$A(J1, J2) = \frac{A+D}{A+B+C+D}$$

In the numbers for systems, the average $A_{avg}(SYSTEM)$ is computed as follows (assuming three human judges):

$$A_{avg}(SYSTEM) = \frac{\sum_{i=1}^{3} A(SYSTEM, J_i)}{3}$$

In the figures for humans, the average $A(H)$ is computed as follows:

$$A_{avg}(H) = \frac{A(J1, J2) + A(J2, J3) + A(J1, J3)}{3}$$

Percentage agreement is problematic as it overestimates the influence of the irrelevant sentences, which are in most cases the larger part of the document (particularly for very short summaries which interest us most here). Precision and recall remedy this shortcoming.

5.1.2 Precision and recall

Precision and Recall are defined as

$$P_{J2}(J1) = \frac{A}{A+C}$$

$$R_{J2}(J1) = \frac{A}{A+B}$$

In our case, each set of documents which is compared has the same number of sentences extracted; thus precision and recall have the same numerical value.

The average $P_{avg}(H)$ and $P_{avg}(SYSTEM)$ are calculated as follows:

$$P_{avg}(SYSTEM) = \frac{\sum_{i=1}^{3} P_J(SYSTEM)}{3}$$

$$P_{avg}(H) = \frac{P_{J2}(J1) + P_{J3}(J2) + P_{J1}(J3)}{3}$$

The averages $R_{avg}(H)$ and $R_{avg}(SYSTEM)$ are calculated correspondingly.

However, both precision and recall (and also percent agreement) do not take chance agreement into account. The amount of agreement one would expect two judges to reach by chance depends on the number and relative proportions of the categories used by the coders. The next section, for instance, shows that chance agreement is very high in our data set.

5.1.3 Kappa

Kappa (Siegel and Castellan, 1988) is a statistical measure which addresses the problem of random agreement. It is increasingly used in NLP annotation work (Krippendorff, 1980; Carletta, 1996). Kappa has the following advantages:

- It factors out random agreement. Random agreement is defined as the level of agreement which would be reached by random annotation using the same distribution of categories as the real annotators.
- It allows for comparisons between arbitrary numbers of annotators and items.
- It treats less frequent categories as more important (in our case: selected sentences), similarly to precision and recall but it also considers (with a smaller weight) more frequent categories as well.

The Kappa coefficient controls agreement $P(A)$ by taking into account agreement by chance $P(E)$:

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

No matter how many items or annotators, or how the categories are distributed, $K = 0$ when there is no agreement other than what would be expected by chance, and $K = 1$ when agreement is perfect. If two annotators agree *less* than expected by chance, Kappa can also be negative.

Kappa is stricter than percent agreement: its value is always lower than or equal to percent agreement P(A); it is equal in the case of a uniform distribution and lower for skewed distributions. We already know that our category distribution is very skewed for low compression rates. Therefore, we expect chance agreement to be quite high in our data.

We report Kappa between three annotators in the case of human agreement, and between three humans and a system (i.e. four judges) in a latter section.

5.2 Content-based methods

5.2.1 Restrictions of co-selection evaluation methods

Two summaries written by two human summarizers, for instance, do not in general share identical sentences. In the following example, it is obvious that both sentences, S_1 and S_3, carry the same meaning. An extractive summary should be rewarded in some way for the selection of sentence S_3.

S_1 "The visit of the president of the United States to China"

S_3 "The US president visited China"

Whereas co-selection measures cannot do this, content-based similarity measures can. Recent research has shown how content-based evaluation can be carried out in automatic or semi-automatic fashion (Donaway et al., 2000; Saggion and Lapalme, 2000b; Jones and Paice, 1992; Paice and Oakes, 1999). The similarity measures we consider in this work are word overlap, longest common subsequence, and cosine similarity. Content-based similarity measures have been used in the past to assess machine translation quality Papineni et al. (2001). We have specified and implemented a number of content-based similarity measures that take into account different properties of the text:

5.2.2 Cosine similarity

Cosine similarity is computed using the following formula (Salton, 1988):

$$cos(X,Y) = \frac{\sum x_i * y_i}{\sqrt{\sum (x_i)^2} * \sqrt{\sum (y_i)^2}},$$

where X and Y are text representations based on a vector space.

5.2.3 Unit Overlap

Unit overlap is computed using the following formula:

$$overlap(X,Y) = \frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|},$$

where X and Y are text representations based on sets. Here $\|S\|$ is the size of set S.

5.2.4 Longest Common Subsequence

Longest Common Subsequence is computed using the formula:

$$2 * lcs(X,Y) = length(X) + length(Y) - edit_{di}(X,Y)$$

Where X and Y are representations based on sequences and where $lcs(X,Y)$ is the length of the longest common subsequence between X and Y, $length(X)$ is the length of the string X, and $edit_{di}(X,Y)$ is the minimum number of deletion and insertions needed to transform X into Y (Crochemore and Rytter, 1994).

When we have to compare to texts using the lcs we apply the following formula:

Let Set_1 and Set_2 be the two sets to compare (these are sets of sentences where each sentence is a sequence of unigrams (words or lemmas in our case)):
$lcs(Set_1, Set_2) = \frac{\sum_{s_1 \in Set_1} max_{s_2 \in Set_2} lcs(s_1,s_2) + \sum_{s_2 \in Set_2} max_{s_1 \in Set_1} lcs(s_1,s_2)}{\sum_{s \in Set_1} length(s) + \sum_{s \in Set_2} length(s)}$
Where $lcs(Set_1, Set_2)$ is a pairwise lcs average of the two texts.

5.2.5 Text Representation

We have considered the following features to represent the content of target and automatic summaries because each measures works on a different structures:

- Sentence Structure: set, vector, or sequence;
- Granularity: unigram or bigram for English and word or character for Chinese;
- Unit: words or lemmas;
- Part of Speech: main verbs, nouns, or all part of speech.

Different representations are obtained combining the features and values here identified. So, for example, the text "The president visited China" will be represented with a set {president, china } if we decided to use the features: set, unigrams, lemmas, and nouns, and will be represented with the sequence "the president visit china" if we decided to use the features: sequence, unigrams, lemmas, and all part of speech. In the case of texts in Chinese, we don't rely on parts of speech (because they were unavailable during the workshop), but we do explore words and Chinese characters as possibilities, because we have developed algorithms to deal with these two text representations. In the case of vector of terms, we use two possible weighting schemes for the terms: presence/absence of the term in the text or $tf * idf$ computed using corpus and within text term distribution.

Cosine similarity uses the vector space with words or lemmas as text representation; unit overlap uses sets of words or lemmas as text representation, and unigrams or bigrams; and longest common subsequence uses sequences of words or lemmas as text representation.

5.3 Relative Utility

We will now discuss Relative Utility (RU), a method for evaluating extractive summarizers, both single-document and multi-document. We will address some advantages of RU over existing co-selection metrics such as precision, recall, percent agreement, and Kappa. We will present some experiments performed on a large text corpus to discuss how RU is affected by interjudge agreement, compression rate (or summary length), and summarization method.

The main problem with Precision, Recall, and Percent Agreement for evaluating extractive summarizers is that human judges often disagree what are the top n% most important sentences in a document or cluster and yet, there appears to be an implicit salience value for all sentences which is judge-independent.

5.3.1 The relative utility evaluation method

Using metrics such as precision and recall (P&R) or percent agreement (PA) (Jing et al., 1998; Goldstein et al., 1999) to evaluate summaries creates the possibility that two equally good extracts are judged very differently.

Suppose that a manual summary contains sentences [1 2] from a document. Suppose also that two systems, A and B, produce summaries consisting of sentences [1 2] and [1 3], respectively. Using P&R or PA, system A will be ranked much higher than system B. It is quite possible, however that for the purpose of summarization, sentences 2 and 3 are equally important, in which case the two systems should get the same score.

The relative utility (RU) method (Radev et al., 2000) allows ideal summaries to consist of sentences with *variable membership*. With RU, the ideal summary represents all sentences of the input document(s) with confidence values for their inclusion in the summary.

For example, a document with five sentences [1 2 3 4 5] is represented as [1/10 2/9 3/9 4/2 5/4]. The second number in each pair indicates the degree to which the given sentence should be part of the summary according to a human judge. We call this number the *utility* of the sentence. Utility depends on the input documents, the summary length, and the judge. In the example, the system that selects sentences [1 2] will not get a higher score than a system that chooses sentences [1 3] given that both summaries [1 2] and [1 3] carry the same number of utility points (10+9). Given that no other combination of two sentences carries a higher utility, both systems [1 2] and [1 3] produce optimal extracts.

5.3.2 An example

In relative utility experiments, judges are asked to assign numerical scores to individual sentences from a single document or a cluster of related documents. A score of 10 indicates that a sentence is central to the topic of the cluster while a score of 0 marks a totally irrelevant sentence.

The following example illustrates an advantage that Relative Utility has over Precision/Recall. The two summaries shown in Figures 5.2 and 5.3 are 5-sentence extractive summaries from the same document by two different judges. Because each summary is composed entirely of different sentences, the interjudge agreement as measured by Precision/Recall is 0, despite the fact that both are reasonable summaries. Both extractive summaries are based on document D-19971207-001 from cluster 398.

S#	Text	J_1 util (of 10)	J_2 util (of 10)
2	The preliminary investigations showed that at this stage, human-to-human transmission of the H5N1 influenza A virus has not been proven and further investigations will be made to study this possibility, the Special Working Group on H5N1 announced today (Sunday)	9	8
3	The initial findings also showed that the four H5 cases did not share a common source, nor was the virus transmitted from one case to the others.	7	4
7	However, there is no cause for panic as available evidence does not suggest that the disease is widespread.	7	6
9	The WHO has been asked to alert vaccine production centres in the world in the case investigation to follow developments here with a view to preparing the necessary vaccines.	7	7
14	He said the Department would disseminate to doctors, medical professionals, colleges and health care workers available information about the H5 virus through letters and the Department of Health's homepage on the Internet (http:/www.info.gov.hk/dh/).	8	8

Figure 5.2: A 5-sentence extractive summary by LDC Judge J_1

S#	Text	J_1 util (of 10)	J_2 util (of 10)
11	To further enhance surveillance in Hong Kong, Dr Saw said, the Department of Health would extend surveillance coverage to all General Out-patient Clinics.	8	10
12	The Hospital Authority would also set up surveillance in public hospitals.	4	10
13	In the meantime, Dr Saw said, the Agriculture and Fisheries Department had also increased surveillance in poultry in collaboration with The University of Hong Kong.	6	10
19	Dr Saw advised members of the public that the best way to combat influenza infection was to build up body resistance by having a proper diet with adequate exercise and rest.	7	10
20	Good ventilation should be maintained to avoid the spread of respiratory tract infection.	8	10

Figure 5.3: A 5-sentence extractive summary by LDC Judge J_2

Note that both judges gave each other's sentences fairly high utility scores, however. In fact, the interjudge agreement as measured by Relative Utility for this example is 0.76. This score is also markedly higher than the lowest possible score a summarizer could receive. Although not depicted above, a summarizer could have an agreement with judge J_1 as low as 0.14 and an agreement with judge J_2 as low as 0.38.

5.3.3 Defining Relative Utility

In this section, we will formally define relative utility. To compute relative utility, a number of judges, N ($N \geq 1$), are asked to assign *utility scores* to all n sentences in a cluster of documents (which can consist of one or more documents). The top e sentences according to utility score are then called a sentence extract of size e (in the case of ties, some arbitrary but consistent mechanism is used to decide which sentences should be included in the summary). The formulas below assume that n is the number of sentences in a cluster of documents, e is the number of sentences in the desired extract, and N is the number of human judges providing utility scores.

We can then define the following metrics:

$$\begin{aligned}
\vec{U_i} &= \{u_{i,1}, u_{i,2}, ..., u_{i,n}\} \\
&= \text{ sentence utility scores for judge } i \\
\vec{U_i'} &= \{\delta_{i,1} \cdot u_{i,1}, \delta_{i,2} \cdot u_{i,2}, ..., \delta_{i,n} \cdot u_{i,n}\} \\
&= \text{ extractive utility scores for judge } i
\end{aligned}$$

In the formula for $\vec{U_i'}$, $\delta_{i,j}$ is the summary characteristic function for judge i and sentence j. It is equal to 1 for the e highest-utility sentences for a given judge. Note that $\sum_{j=1}^{n} \delta_{i,j} = e$.

$$\delta_{i,j} = \begin{cases} 1 & \text{sentence } j \text{ is included in the extract} \\ & \text{judge } i \text{ at a given summary length } e \\ 0 & \text{otherwise} \end{cases}$$

We can now define some additional quantities:

$$
\begin{aligned}
U_i &= \sum_{j=1}^{n} u_{i,j} \\
&= \text{total self-utility for judge } i \\
U'_i &= \sum_{j=1}^{n} \delta_{i,j} \cdot u_{i,j} \\
&= \text{total extractive self-utility for judge } i \\
&\quad \text{(computed over all } n \text{ sentences)} \\
U_{i,k} &= \sum_{j=1}^{n} \delta_{i,j} \cdot u_{k,j} \\
&= \text{total extractive cross-utility for judges } i \text{ and } k \ (i \neq k) \\
U_{i,avg} &= 1/(N-1) \cdot \sum_{k=1}^{N} U_{i,k} \quad \text{for } i \neq k \\
&= \text{(non-symmetric) judge utility for judge } i. \\
J &= U_{avg} = 1/N \cdot \sum_{i=1}^{N} U_{i,avg} \\
&= \text{interjudge performance} \\
&= \text{average extractive cross-utility of all judges.} \\
U &= \sum_{j=1}^{n} \sum_{i=1}^{N} u_{i,j} \\
&= \text{total extractive utility for all judges.} \\
U' &= \sum_{j=1}^{n} \varepsilon_j \cdot \sum_{i=1}^{N} u_{i,j} \\
&= \text{total utility for all judges}
\end{aligned}
$$

In the formula for U', ε_j is 1 for the top e sentences according to the sum of utility scores from all judges. U' is the maximum utility that any system can achieve at a given summary length e.

Note that $\sum_{j=1}^{n} \varepsilon_{i,j} = e$. Note also that $N = 1$ implies $U' = U'_1$ (single judge case).

A summarizer producing an extract of length e can be thought of as an additional judge. Its performance will be computed as the ratio of the sum of its cross-utility with the totality of human judges to the maximum utility U' achievable at a given summary length e. As a result, a summary can be judged based on its utility *relative* to the maximum possible against the set of judges, hence the name of the method *Relative Utility*.

$$
\begin{aligned}
S &= \frac{\sum_{j=1}^{n} \delta_{s,j} \cdot \sum_{i=1}^{N} u_{i,j}}{U'} \\
&= \text{system performance } (\delta_{s,j} \text{ is equal to 1 for the} \\
&\quad \text{top } e \text{ sentences extracted by the system).}
\end{aligned}
$$

In the formula for S, $\sum_{i=1}^{N} u_{i,j}$ is the utility assigned by the totality of judges to a given sentence j extracted by the summarizer.

$$
\begin{aligned}
R &= 1/\binom{n}{e} \sum_{t=1}^{\binom{n}{e}} S_t \\
&= \text{random performance (computed over all } \binom{n}{e} \\
&\quad \text{possible extracts of length } e).
\end{aligned}
$$

R is practically a lower bound on S while J is the corresponding upper bound. In order to factor in the difficulty of a given cluster, one can normalize the system performance S between J and R:

$$\begin{aligned} D &= \frac{S-R}{J-R} \\ &= \text{normalized relative utility} \\ &\quad \text{(normalized system performance).} \end{aligned}$$

Assuming $R \neq J$ (which is a reasonable assumption), $D = 1$ only when $S = J$ (system is as good as the interjudge agreement) and $D = 0$ when $S = R$ (system is no better than random).

When values for R and J are given as comparison, reporting S is sufficient. However, D should be used when R and J are ignored.

Given that Relative Utility S values are generally higher than Kappa values, one might incorrectly think that system performance is quite high. Although RU does indeed take into account variable agreement, one should be aware that normalized Relative Utility (D) rather than S takes random and interjudge agreement into account. It is also important to note that even D values are not directly comparable with Kappa. Further work is required to establish more solid statistical properties of RU.

5.3.4 Comparing Relative Utility with P/R

To understand this section better, please refer to Figure 2.14.

Given an *ideal* extract E_1 consisting of e_1 sentences, one can measure how similar another extract E_2 including e_2 sentences is to it. Precision (P) is the ratio of sentences included in E_2 which are also included in E_1 while Recall (R) is the ratio of sentences included in E_2 to the total number e_1 of sentences in E_1. It can be trivially shown that if $e_1 = e_2 = e$ and the two extracts have a sentences in common, $P = R = a/e$.

Percent agreement (PA) measures how many of the judges' decisions are shared amongst two judges. If d is the number of sentences in the input document (or cluster) that were not extracted by either judge and the input has n sentences, then PA is defined as $(a + d)/n$.

For example, suppose that two judges produce 10% extracts from a document containing 50 sentences. If 4 sentences are extracted by both judges, then $P = R = 4/5 = 80\%; PA = (4 + 44)/50 = 96\%$. PA is known to significantly overestimate agreement for both very short and very long extracts while P and R underestimate agreement.

We can now compare the RU values with these for Precision and Recall. Let's first look at judges 1 and 2. Out of 24 sentences, only 4 overlap between the two judges (19980306_007:2, 19990802_006:8, 19990802_006:9, and 19990829_012:2), or in other words, $P = R = 4/24 = .17$. (Note that when the two extracts are of the same length, Precision trivially equals Recall). Let's now look at judges 1 and 3. They overlap on only 3 sentences ($P = R = .13$). Similarly, $P = R = .13$ for judges 2 and 3.

Let's now turn to the performance of MEAD. MEAD has $P = R = 2/24 = .08$ with judge 1. The values for P and R are .13 and .17 when comparing MEAD with judge 2 and judge 3, respectively.

Such low numbers could indicate that it is impossible to reach consensus on extractive summaries. The numbers above are for multi-document extracts, although similar numbers hold for single-document extracts as well. For example, the average interjudge P/R for 10% extracts of each of the ten single documents comprising cluster 125 is .22 for judges 1 and 2, .33 for judges 2 and 3, and .26 for judges 3 and 1.

Past work on evaluating extractive summaries (Jing et al., 1998; Goldstein et al., 1999) has indicated such low agreement for single-document extracts. We claim that Relative Utility is a better metric than P/R because it doesn't underestimate agreement in the case where multiple sentences are almost equally good to be included in an extract.

Relative Utility has several additional advantages over P/R/PA.

First, in a way similar to Kappa (Siegel and Castellan, 1988), it takes into account the difficulty of a problem by factoring in random and interjudge performance.

Second (and unlike Kappa), it can be used for evaluation at multiple compression rates (summary lengths). In one pass, judges assign salience scores to all sentences in a cluster (or in a single document). It is then possible to simulate extraction at a fixed compression rate by ranking sentence by utility. As a result, RU is a more informative measure of sentence salience.

Third, RU can be extended to deal with informational subsumption by introducing conditional sentence utility values (Radev et al., 2000) which depend on the presence of other sentences in the summary. Informational subsumption deals with the fact that the utility of a sentence may depend on the utility of other sentences already included in a summary. For example, two sentences may be almost identical in content and get the same utility scores from a judge and yet they should not be included in the summary at the same time. In this report, we are not presenting any results related to subsumption although we obtained subsumption data for the 20-cluster corpus. We intend to use this raw data for future experiments.

Fourth, the RU method can be further expanded to allow sentences or paragraphs to exert negative reinforcement on one another, that is, allow for cases in which the inclusion of a given sentence makes another redundant and a system that includes both will be penalized more than a system which only includes one of the two "equivalent" sentences and another, perhaps less informative sentence.

5.3.5 Extracts

An extract contains a list of sentences that will be used in the summary. Sentences are sorted in the order they appear.

We used MEAD to produce a large number of automatic extracts (at 10 target lengths using a number of algorithms of all 20 clusters and of all 18,146 documents in the corpus).

Figure 2.14 presents seven different 10% extracts produced from the same cluster (Cluster 125). As one can see, when all judges are taken into account, one sentence with high salience is sentence 2 from article 19980306_007 with a total utility score of 24. Given that MEAD includes that sentence in its 10% extract, it will get the maximum possible utility for this sentence. On the other hand, not all sentences extracted by MEAD have this high a utility. For example, sentence 3 from 19990802_006 which was also picked by MEAD only carries a utility of 15. If MEAD had picked a different sentence instead (e.g., sentence 2 from 20000408_011 with a utility of 28), its relative utility would be higher.

In this example, the total self-utility U_1 for judge 1 is 1218. The total self-utilities for judges 2 and 3 are 1380 and 1130, respectively. The values for extractive total utility $U_i^!$ for each of the three judges are 237, 218, and 224, respectively.

Figure 5.4 shows the values for extractive cross-judge utility. The average, 0.73, is equal to the interjudge agreement J.

	Judge 1	Judge 2	Judge 3	Average
Judge 1	1.00	0.74	0.74	0.74
Judge 2	0.64	1.00	0.74	0.69
Judge 3	0.72	0.81	1.00	0.77

Figure 5.4: Cross-judge utilities

Using the formulas in the previous section, one can compute the value for random performance, which is 0.57.

The performance of MEAD is 0.70 (compared to random = 0.57 and interjudge agreement = 0.73). When normalized, MEAD's performance is 0.80 on a scale from 0 to 1.

5.4 IR Evaluation Measures

Relevance correlation which will be introduced in more detail in the following section requires a measurement of IR performance as its backdrop. We will discuss here several IR techniques known in the literature.

Traditionally, evaluation of retrieval performance is measured by *recall* and *precision*. Consider a particular query and a set of known relevant documents (gold standard). Suppose a retrieval system returns a set of system-determined relevant documents. In general, retrieval effectiveness can be depicted in the following table:

	Truth		
	Relevant	Non-relevant	
Retrieved	A	B	$A+B$
Non-retrieved	C	D	$C+D$
	$A+C$	$B+D$	$N=A+B+C+D$

Recall and precision can be easily computed using this table. Specifically, recall is defined as the fraction of the relevant documents which has been retrieved. It can be expressed as:

$$\text{Recall} = A/(A + C)$$

Precision is defined as the fraction of the retrieved documents which is relevant. It can be expressed as:

$$\text{Precision} = A/(A + B)$$

Sometimes, a system returns a ranked list of documents sorted by degree of relevance instead of just a set of relevant documents. We can then examine this ranked list starting from the top. The precision and recall are computed after each relevant document encountered in the list. This can produce a recall-precision graph which plots precision as a function of recall. This graph also depicts the behavior of a retrieval run over the entire recall spectrum. Normally precision and recall tend to be inversely related. *Non-interpolated Average Precision* is defined as the average of the precision obtained at various recall levels obtained. The graph can also allow us to compute *11-point Average Precision*. 11-point Average Precision is defined as the average of the precision obtained at 11 standard recall levels which are 0%, 10%, ... , 100%. Interpolation procedure is applied on the recall-precision curve in order to obtain the precision of these standard recall levels.

In the above discussion, the evaluation is performed on a single query. In order to get a more reliable evaluation, a number of test queries are used. Basically, we first calculate the measures discussed above of each query and take the mean over all test queries. Hence, *Mean (non-interpolated) Average Precision* for a run is the mean of the Average Precision of each query.

There are other measures which attempt to compute average precision at given document cutoff values. For example, P(10) is the average precision after the first 10 documents are retrieved. Likewise, R(1000) is the average recall after the first 1000 documents are retrieved.

The `trec_eval` program written by NIST produces the evaluation measures discussed above. This program can be obtained from the TREC Web site (http://trec.nist.gov).

5.5 Evaluation Framework for Chinese Summaries

The experimental framework for evaluation of the Chinese summaries is based on the novel idea of using the aligned corpus as a source for obtaining target abstract in Chinese. The framework is shown in Figure 5.5. The steps involved in the evaluation are:

(1) LDC provided sentence utility judgments between 0 and 10 for each English document in the clusters. Each document was judges by three different assessors;

(2) the utility-based summarizer was run on each document for different compression rates;

(3) the utility-based summaries were mapped into Chinese using the table of sentence alignments;

(4) Different summarizers were used to produce single and multi document summaries in Chinese at different compression rates;

(5) Content-based functions were used to compute the similarity between the texts.

5.6 Relevance Correlation

Relevance correlation is a new measure for assessing the relative decrease in retrieval performance when moving from full documents to extracts. To our knowledge, this measurement has never been explored in detail, and certainly never on such a large data set as ours. Relevance correlation takes the *absolute* retrieval results reported in section 6.4 and turns them into a *relative* measure of summary performance. The idea behind it is as follows: if a summary captures the main points of a document, then an IR machine indexed on a set of such summaries (instead of a set of the full documents) should produce (almost) as good a result. Moreover, the difference between how well the summaries do and how well the full documents do should serve as a possible measure for the quality of summaries.

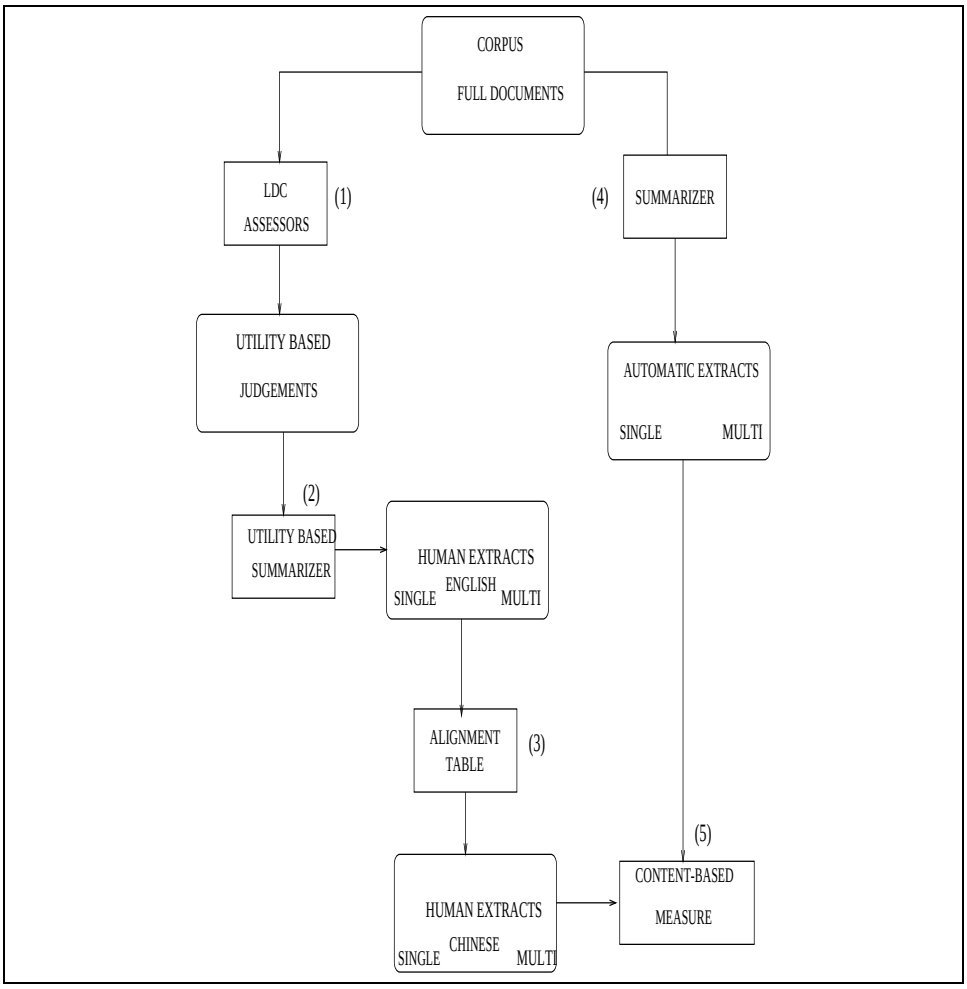


Figure 5.5: Evaluation Framework (Chinese)

Brandow et al. (1995) suggest a similar evaluation measure to measure the relative effectiveness of their summarizer ANES in comparison to leading-text summaries. In their experiment, 12 Boolean queries are run on a test corpus of roughly 21,000 documents. Relevance judgements were collected and compared for all retrieved documents. In three conditions, the documents which are indexed are either full documents, ANES extracts, or lead extracts.

In their results, leading text outperformed full text, and leading text also outperformed ANES extracts in precision (47% vs. 45% vs. 37%). Recall however is 100% on full documents, and 56% (ANES) vs. 58% (leading). In their future work section, they note that it would be important to repeat these experiments with a non-Boolean search engine.

Our work does just that, and it measures retrieval performance with more measurements and more conditions, on a larger set of queries. Our results are thus more general.

Suppose that given a query Q and a corpus of documents D_i, a search engine ranks all documents in D_i according to their relevance to the query Q. If instead of the corpus D_i, the respective summaries of all documents are substituted for the full documents and the resulting corpus of summaries S_i is ranked by the same retrieval engine for relevance to the query, a different ranking will be obtained. If the summaries are good surrogates for the full documents, then it can be expected that ranking will be similar. There exist several methods for measuring the similarity of rankings. One such method is Kendall's tau and another is Spearman's rank correlation. Both methods are quite appropriate for the task that we want to perform, however, since search engines produce relevance scores in addition to rankings, we can use a stronger similarity test, linear correlation. When two identical

rankings are compared, their correlation is 1. Two completely independent rankings result in a score of 0 while two rankings that are reverse versions of one another have a score of -1.

Relevance correlation r is defined as the linear correlation of the relevance scores (x and y) assigned by two different IR algorithms on the same set of documents or by the same IR algorithm on different data sets. Relevance scores are obtained using each of the 20 queries described in 2.7.

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2}\sqrt{\sum_i (y_i - \overline{y})^2}}$$

A standard vector-space model (Salton and McGill, 1983) is employed as our retrieval method. Each summary is represented as a k-dimensional vector where each element in the vector denotes the weight of the corresponding term. Likewise, a query is represented in vector form. For English summaries, we apply stemming and stop-word removal to extract terms from the summaries for indexing. For Chinese summaries, we first process the texts by a word segmentation step to detect word boundaries in a similar way as the full-length Chinese documents.

All English and Chinese summaries in the whole corpus are indexed, then retrieval is performed.

After the retrieval process, each summary is associated with a score indicating the relevance of the summary to the query. The relevance score is actually calculated as a cross product between the summary vector and the query vector. Based on the relevance score, we can produce a full ranking of all the summaries in the corpus. This ranking of summaries is stored in an XML file called the `docjudge` file in descending order of relevance score. Different sets of `docjudge` files are generated for different sets of summaries. A separate set of `docjudge` files is produced for the original corpus of the full-length documents.

In contrast to Brandow et al. (1995) who run 12 Boolean queries on a corpus of 21,000 documents and compare three types of documents (full documents, lead extracts, and ANES extracts), we measure retrieval performance under more than 300 conditions (by language, summary length, retrieval policy for 8 summarizers or baselines). Our results are thus more general.

Chapter 6

Results

In this chapter, we will present our results obtained using the five types of evaluation measures introduced in Chapter 3. The five types of metrics are (1) co-selection, (2) content-based, (3) relative utility, (4) information retrieval, and (5) relevance correlation.

6.1 Co-selection results

Co-selection agreement is reported using the three evaluation measures discussed in Section 5.1: percent agreement, precision and recall, and kappa.

An overview of the results (averaged over the 20 development clusters) is given in Figures 6.2, 6.4 and 6.7. The tables assume human performance is the upper bound, the next two rows treat MEAD and WebSumm, the two systems considered, and the lower two lines consider the baselines.

6.1.1 Percent agreement

The first row in figure 6.2 and the graph show agreement amongst the human annotators if measured in percent agreement. These figures are given for an average of 20 development clusters (detailed results by cluster can be found in the appendix).

Numbers for humans show a comparison of three extracts. Numbers for all other "systems", i.e. random, lead-based, MEAD and WebSumm show agreement of the system against the three annotators, i.e. instead of comparing three extracts, we compare *four*.

Notice that the values are higher in the extreme compressions and lower in the mid-range compressions. This is due to the fact that percent agreement is a very crude measure that does not take into account the distribution of the classification. Particularly in the lower and higher compressions, the distribution is very skewed, as most of the sentences in a document are either irrelevant, or they are relevant in these two situations. But percent agreement does not take the skewedness of the distribution into account; each sentence is considered equally important, whether it is relevant or not. As discussed earlier, this undesirable effect is due to the insensitivity of the evaluation measure to random agreement.

It is also the case that percent agreement is not robust towards number of annotators. If we plot the agreement of several summarization strategies (random, lead-based, MEAD and WebSummn[1]) with all three human summarizers (cf. figures 6.11, 6.12, 6.13 and 6.14), we note that the numbers are lower than those of the three humans compared to each other.

Random agreement is the lowest baseline, which is met by all systems and the humans. But the comparison of three humans does not achieve the highest agreement in this dataset: lead summaries seem to perform better than the more sophisticated summarizers, and they also seem to be more "similar" to the humans than the humans are amongst themselves! While this result might seem counterintuitive, the observed effect could be explained by a scenario in which human judges choose different, but early occurring sentences. All three humans represent three different points in the solution-space of the extraction problem. It is then possible that one solution (be it a

[1]Only summaries for compression rates up to 40% were available.

	Target Length									
	5%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Humans	.883	.822	.723	.661	.646	.651	.682	.734	.810	.907
MEAD	.880	.818	.714	.647	.625	.626	.658	.715	.791	.897
WebSumm	.884	.816	.707	.633	.607					
Lead	.890	.831	.730	.659	.640	.642	.665	.721	.799	.903
Random	.874	.806	.690	.614	.580	.572	.609	.668	.757	.884

Figure 6.1: Results in percent agreement for all systems, averaged over 20 queries

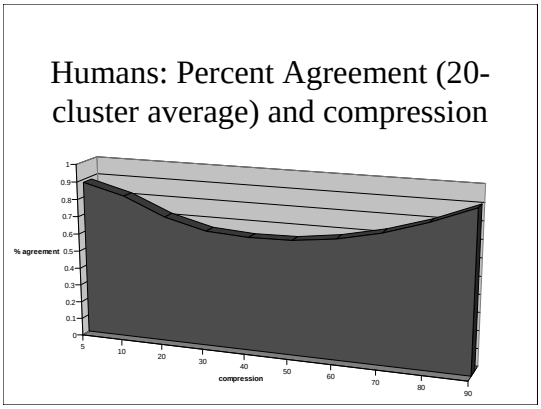


Figure 6.2: Results in percent agreement for humans, averaged over 20 queries

	Target Length									
	5%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Humans	.187	.246	.379	.467	.579	.672	.773	.847	.906	.957
MEAD	.160	.231	.351	.420	.519	.611	.723	.807	.871	.952
WebSumm	.310	.305	.358	.439	.543					
Lead	.354	.387	.447	.483	.583	.652	.726	.818	.888	.954
Random	.094	.113	.224	.357	.432	.518	.638	.734	.834	.939

Figure 6.3: Results in precision=recall for all systems, averaged over 20 queries

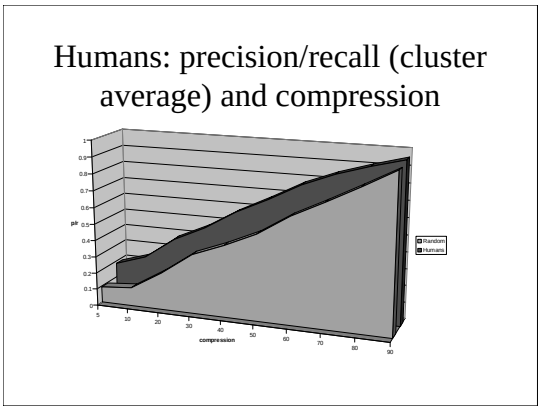


Figure 6.4: Results in precision=recall for humans, averaged over 20 queries

	Target Length									
	5%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
Humans	.127	.157	.194	.225	.274	.302	.323	.337	.354	.324
MEAD	.109	.136	.168	.192	.230	.252	.274	.290	.290	.253
WebSumm	.138	.128	.146	.159	.192					
Lead	.180	.198	.213	.220	.261	.284	.287	.304	.316	.300
Random	.064	.081	.097	.116	.137	.145	.169	.171	.175	.156

Figure 6.5: Results in kappa for all systems, averaged over 20 queries

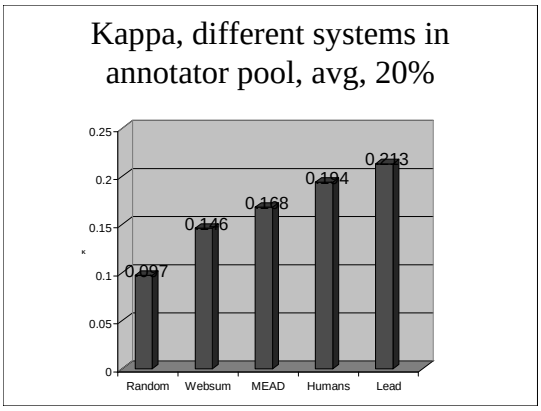


Figure 6.6: Averages in kappa for all systems at compression of 20%

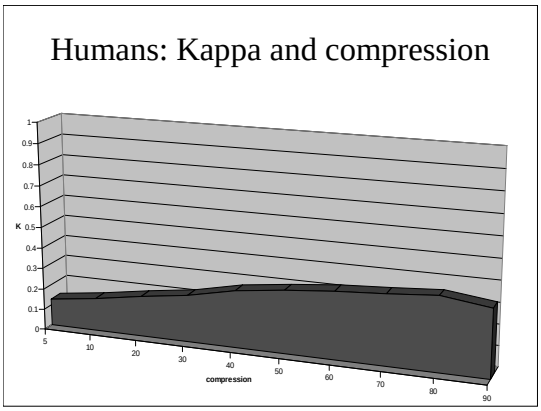


Figure 6.7: Results in kappa for humans, averaged over 20 queries

baseline or a system, or as in our case, lead-based summaries) positions itself between the humans. While humans do not necessarily agree with each other, they seem to agree on the strategy of choosing sentences occurring early in the document.

When comparing the different systems, Websumm achieves better results in the extreme high-target length case than MEAD, but all other compressions show an advantage for MEAD.

6.1.2 Precision and recall

A clearer picture of system performance emerges from Figure 6.4, as precision and recall treat only relevant items. Please note that the extracts we compare have exactly the same number of sentences. Therefore, precision and recall are numerically equal; we only present one value for both of them.

We observe the effect of a dependence of the numerical results on the length, which is a well-known fact from information retrieval evaluations. We also see that random agreement still has a negative effect on the numbers.

Figure 6.15 shows precision/recall values for the three human annotators. The effect of a dent in the mid-range, which we observed with percent agreement, has disappeared, values increase steadily with compression.

With respect to comparing systems, Websumm again has an advantage over MEAD for longer summaries but not for 20% or less. Humans only catch up with lead-based summaries in lengths of 50% or more.

6.1.3 Kappa

Kappa is a superior measure to both precision and recall and to percent agreement as it factors random agreement out. Figure 6.7 summarize results for all compression rates and systems, averaged over clusters. Again, more detailed tables (by cluster) can be found in the appendix.

The numerical figures in Kappa are lower than percent agreement, which is a side-effect of a random agreement larger than 0. The rather large numerical difference between the numbers in Figures 6.2 and 6.7 show that random agreement is rather high in our data set.

One baseline is not shown here, but is built into the definition of Kappa to be zero: comparion of *one* human with a random process whereby the distribution is followed is zero on average. We have empirically confirmed with experiments (not shown here) that random agreement does achieve K=0 if we compare the random processor against a human (or any other process).

The interpretation of Kappa values is possible according to two scales. On Krippendorff's (1980) scale, agreement of K=.8 or above is considered as reliable, agreement of .67–.8 as marginally reliable and agreement of K<.67 as unreliable. On Landis and Koch's (1977) more forgiving scale, agreement of .0–.2 is considered as showing "slight" correlation, .21–.4 as "fair", .41–.6 as "moderate", .61–0.8 as "substantial", and .81 –1.0 as "almost perfect".

Our results, while very different from random, do not show high agreement amongst humans in our case. The relatively low agreement can be due to a number of reasons. Firstly, the literature has long noted low human agreement in the task of sentence extraction. It simply does not seem to be a problem that is intuitive to humans, given the vagueness of relevance in general. Secondly, if a task is not intrinsically intuitive, one can still achieve high agreement by training humans and by cyclically improving the guidelines, which are the guard of the semantics of each decision. In our case, we had given the annotators only vague guidelines; we had no way of making those guidelines more stringent after measuring the first results, like one would normally do, and we had in fact no second chance of training the annotators again. In the light of these restrictions, and considering that sentence extraction is a low-agreement task, the agreement achieved can be considered reasonable.

The numbers nevertheless show the following trends:

- The numbers also show a rather low agreement between humans and all systems and baselines. However, the automatic systems beat the random baseline by far.
- MEAD outperforms Websumm for all but the 5% target length.
- Lead summaries perform best below 20%, whereas human agreement is higher after that.
- There is a rather large difference between the two summarizers and the humans (except for the 5% case for Websumm). This numerical difference is relatively higher than for any other co-selection measure treated here.

	Cluster									
Totals	112	125	199	241	323	398	551	883	1014	1197
Humans	.197	.454	.232	.343	.033	.301	.389	.190	.667	0.395
Random vs. Humans	.125	.356	.211	.247	.089	.200	.294	.169	.500	.270

Figure 6.8: Totals, humans vs. random multidocument extraction, kappa, 10 clusters

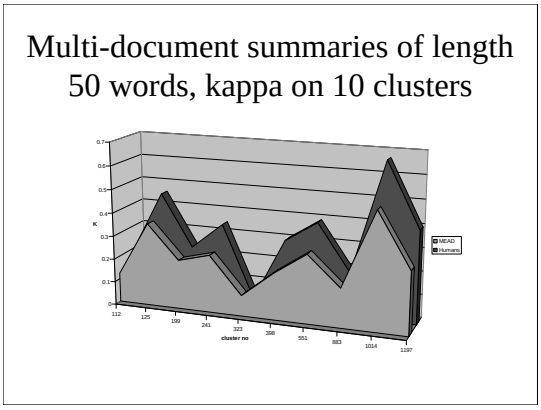


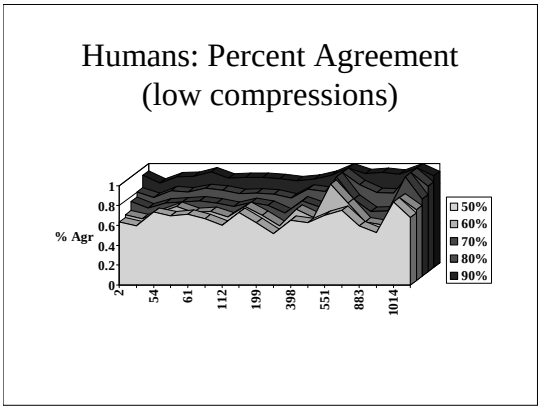
Figure 6.9: Multi-document results (50 words, kappa, 10 clusters)

- Agreement improves with summary length.

We also have some results on multi-document summaries (note that all results quoted above concerned single-document summaries). Figure 6.8 and figure 6.9 show these results for the first 10 clusters. Similarly to the single-document results, these results show large differences between clusters. We report the multidocument results in kappa only, as kappa has the best properties of all co-selection measures considered.

One of our conclusions must be that co-selection is not one of the most sensitive metrics for summarization system performance, although it is commonly used in the field. As the next section will show, similarity measures are a much better metric of similarity between ideal summary and system summary.

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.881	.801	.678	.617	.609	.632	.663	.751	.805	.927
46	.892	.818	.695	.623	.590	.594	.623	.687	.749	.848
54	.881	.860	.817	.788	.720	.732	.686	.737	.813	.911
60	.891	.828	.754	.719	.692	.696	.747	.747	.801	.914
61	.867	.822	.739	.756	.744	.711	.683	.756	.839	.961
62	.916	.836	.762	.695	.663	.665	.695	.735	.817	.898
112	.871	.803	.693	.614	.621	.598	.640	.701	.788	.902
125	.876	.830	.750	.718	.721	.727	.744	.739	.790	.902
199	.891	.833	.708	.630	.611	.627	.659	.714	.779	.891
323	.873	.766	.669	.608	.532	.517	.552	.634	.740	.868
398	.885	.809	.667	.598	.618	.651	.717	.759	.828	.885
447	.884	.810	.697	.653	.639	.629	.633	.697	.806	.925
551	.812	.812	.634	.531	.531	.700	.972	1.000	1.000	1.000
827	.868	.817	.740	.679	.720	.751	.781	.776	.873	.944
883	.863	.803	.716	.634	.585	.596	.590	.650	.798	.956
885	.865	.838	.710	.650	.569	.529	.562	.657	.798	.939
1014	.833	.833	.833	.667	.833	.833	.833	1.000	1.000	1.000
1197	.900	.848	.811	.748	.725	.680	.714	.777	.860	.923
241	.884	.802	.706	.648	.641	.672	.723	.798	.870	.897
1018	.876	.805	.700	.637	.607	.633	.667	.704	.787	.891
TOTAL	.883	.822	.723	.661	.646	.651	.682	.734	.810	.907



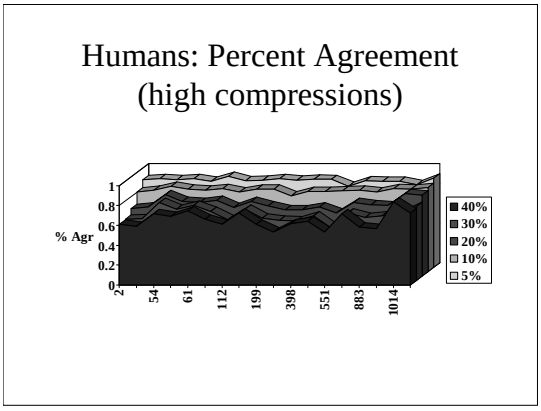


Figure 6.10: Agreement between 3 human annotators, percentage agreement

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.883	.789	.667	.577	.557	.565	.609	.659	.749	.891
46	.890	.818	.692	.601	.558	.547	.568	.637	.721	.840
54	.860	.820	.724	.677	.630	.644	.631	.700	.744	.876
60	.879	.801	.712	.637	.602	.579	.657	.712	.747	.887
61	.833	.800	.672	.669	.622	.600	.603	.692	.783	.942
62	.903	.817	.724	.625	.580	.565	.603	.663	.750	.874
112	.873	.794	.674	.604	.555	.549	.591	.644	.750	.883
125	.879	.813	.693	.655	.615	.612	.629	.672	.753	.878
199	.888	.821	.682	.599	.560	.543	.611	.654	.734	.861
323	.865	.753	.659	.601	.534	.509	.555	.618	.733	.863
398	.882	.806	.664	.576	.557	.561	.599	.669	.764	.863
447	.869	.794	.679	.612	.570	.556	.580	.655	.738	.895
551	.770	.784	.629	.577	.535	.620	.775	.845	.859	1.000
827	.863	.786	.695	.599	.606	.644	.632	.674	.816	.920
883	.866	.798	.678	.590	.563	.546	.571	.623	.776	.926
885	.842	.815	.663	.616	.569	.529	.569	.623	.764	.919
1014	.800	.800	.708	.617	.642	.708	.738	.850	.887	1.000
1197	.886	.814	.738	.641	.634	.584	.637	.668	.788	.884
241	.875	.789	.690	.592	.580	.575	.601	.703	.777	.873
1018	.873	.790	.686	.608	.550	.579	.611	.662	.729	.875
TOTAL	.874	.806	.690	.614	.580	.572	.609	.668	.757	.884

Figure 6.11: Agreement of random summary with 3 human annotators, percent agreement

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.875	.803	.690	.632	.621	.625	.651	.722	.789	.921
46	.910	.844	.726	.657	.632	.645	.670	.725	.782	.867
54	.896	.887	.826	.788	.738	.770	.737	.778	.844	.920
60	.887	.827	.727	.669	.639	.655	.717	.733	.803	.918
61	.850	.814	.706	.694	.692	.669	.694	.781	.836	.950
62	.907	.831	.745	.668	.633	.629	.649	.700	.784	.888
112	.869	.794	.678	.598	.610	.589	.617	.678	.786	.913
125	.886	.836	.741	.688	.695	.707	.713	.727	.796	.904
199	.890	.820	.695	.608	.588	.594	.624	.695	.768	.883
323	.898	.809	.728	.667	.593	.570	.580	.641	.746	.868
398	.900	.829	.690	.594	.577	.585	.614	.666	.749	.862
447	.874	.811	.728	.684	.660	.641	.641	.687	.786	.905
551	.859	.859	.723	.615	.620	.709	.972	1.000	1.000	1.000
827	.870	.789	.718	.637	.654	.677	.709	.735	.831	.930
883	.869	.822	.738	.667	.617	.607	.585	.653	.790	.929
885	.899	.879	.768	.697	.603	.566	.582	.653	.795	.939
1014	.875	.875	.875	.708	.875	.875	.792	1.000	1.000	1.000
1197	.913	.857	.788	.715	.677	.651	.681	.744	.820	.918
241	.885	.813	.685	.601	.585	.601	.637	.721	.815	.898
1018	.882	.803	.699	.651	.632	.661	.671	.705	.784	.900
TOTAL	.890	.831	.730	.659	.640	.642	.665	.721	.799	.903

Figure 6.12: Agreement of lead-based vs. 3 human annotators, percent agreement

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.881	.810	.682	.602	.600	.626	.663	.743	.812	.916
46	.900	.826	.705	.624	.589	.576	.594	.652	.734	.846
54	.858	.805	.730	.699	.661	.651	.637	.687	.772	.874
60	.887	.811	.717	.653	.620	.626	.674	.721	.786	.903
61	.892	.853	.739	.750	.689	.667	.678	.739	.808	.964
62	.908	.838	.745	.688	.646	.634	.677	.726	.810	.898
112	.877	.807	.699	.638	.631	.625	.670	.720	.777	.907
125	.875	.823	.731	.693	.672	.678	.688	.710	.774	.886
199	.892	.829	.717	.631	.604	.611	.649	.720	.783	.886
323	.865	.768	.677	.593	.532	.545	.590	.644	.733	.860
398	.884	.813	.679	.606	.606	.600	.644	.706	.793	.880
447	.874	.803	.701	.662	.650	.636	.645	.716	.801	.889
551	.808	.808	.587	.498	.502	.653	.836	.859	.859	1.000
827	.870	.804	.743	.681	.718	.753	.777	.751	.816	.917
883	.866	.803	.705	.615	.593	.585	.598	.669	.798	.923
885	.835	.808	.687	.646	.582	.562	.576	.633	.761	.919
1014	.825	.825	.833	.633	.750	.750	.800	1.000	1.000	1.000
1197	.884	.821	.757	.684	.647	.635	.681	.740	.824	.918
241	.885	.813	.714	.638	.621	.624	.672	.732	.824	.896
1018	.880	.801	.698	.630	.595	.608	.640	.693	.773	.876
TOTAL	.880	.818	.714	.647	.625	.626	.658	.715	.791	.897

Figure 6.13: Agreement of MEAD vs. 3 human annotators, percent agreement

	Compression				
Clusters	05	10	20	30	40
2	.872	.784	.657	.569	.531
46	.899	.822	.688	.599	.558
54	.881	.839	.788	.764	.711
60	.873	.789	.698	.626	.604
61	.847	.800	.689	.669	.647
62	.902	.815	.719	.633	.594
112	.875	.801	.676	.621	.593
125	.881	.825	.727	.677	.664
199	.889	.822	.694	.612	.572
323	.896	.774	.669	.595	.552
398	.889	.810	.680	.603	.582
447	.869	.794	.692	.639	.612
551	.831	.831	.714	.624	.624
827	.873	.802	.735	.667	.682
883	.874	.814	.713	.604	.546
885	.879	.859	.717	.633	.572
1014	.875	.875	.792	.667	.875
1197	.908	.845	.763	.681	.657
241	.884	.818	.721	.650	.625
1018	.878	.798	.673	.596	.563
TOTAL	.884	.816	.707	.633	.607

Figure 6.14: Agreement of WEBSUMM with 3 human annotators, percent agreement

6.2 Content-based results

We have evaluated summaries for a set of 10 clusters containing 10 documents each. We present the average results over the set of 1000 documents. It is worth mentioning that all content-based similarity measures are more sensitive than co-selection metrics.

6.2.1 Simple Cosine Similarity

The results obtained with these measures for all the representations chosen can be seen in Figures 6.25, 6.26, 6.27, and 6.28. Using this measure, MEAD obtain results close to the human extracts in most of the compression rates.

6.2.2 $tf * idf$ Cosine Similarity

The results obtained with these measures for all the representations chosen can be seen in Figures 6.29, 6.30, 6.31, and 6.32. Using this measure, Lead based obtain results close to the human extracts in most of the compression rates while MEAD is ranked in second position.

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.167	.178	.266	.399	.529	.642	.735	.836	.888	.962
46	.078	.154	.267	.397	.496	.596	.695	.784	.848	.919
54	.233	.378	.581	.656	.675	.740	.752	.837	.906	.959
60	.250	.354	.443	.603	.670	.735	.822	.837	.894	.962
61	.350	.339	.507	.638	.733	.740	.720	.839	.885	.992
62	.255	.228	.391	.481	.552	.632	.717	.791	.873	.939
112	.067	.161	.308	.394	.551	.617	.723	.804	.876	.950
125	.094	.252	.439	.561	.678	.747	.810	.825	.872	.946
199	.127	.268	.314	.413	.519	.616	.704	.790	.858	.932
241	.200	.186	.319	.462	.576	.672	.768	.856	.917	.945
323	.167	.233	.276	.418	.441	.525	.651	.756	.844	.929
398	.094	.153	.223	.357	.548	.671	.772	.825	.887	.932
447	.250	.244	.328	.450	.568	.626	.709	.798	.885	.967
551	.333	.333	.350	.443	.457	.733	.980	1.000	1.000	1.000
827	.133	.300	.400	.516	.679	.755	.825	.852	.930	.974
883	.167	.183	.344	.467	.528	.620	.685	.766	.886	.979
885	.333	.333	.367	.470	.508	.554	.664	.765	.881	.973
1014	.333	.333	.667	.557	.833	.833	.867	1.000	1.000	1.000
1018	.161	.288	.376	.443	.541	.653	.741	.808	.898	.964
1197	.283	.360	.551	.596	.663	.684	.768	.850	.919	.962
TOTAL	.187	.246	.379	.467	.579	.672	.773	.847	.906	.957

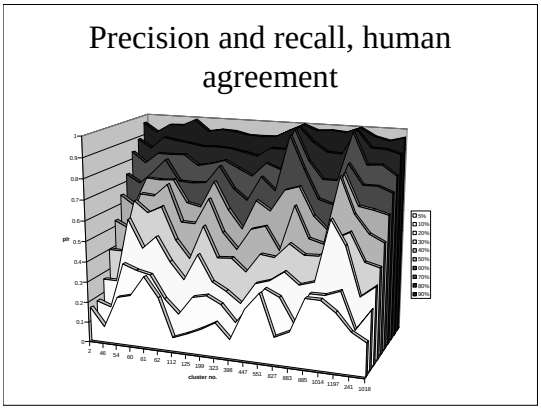


Figure 6.15: Agreement between 3 human annotators in precision (= recall)

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.167	.094	.216	.278	.421	.499	.627	.705	.810	.919
46	.039	.159	.255	.328	.419	.503	.605	.706	.810	.908
54	.033	.144	.173	.304	.454	.549	.652	.781	.803	.915
60	.128	.173	.285	.305	.385	.456	.652	.782	.816	.928
61	.000	.119	.179	.390	.482	.523	.630	.782	.840	.985
62	.011	.067	.250	.282	.376	.477	.605	.720	.807	.921
112	.100	.089	.262	.376	.399	.506	.630	.722	.831	.930
125	.144	.115	.156	.367	.401	.508	.608	.728	.826	.922
199	.092	.174	.165	.289	.415	.471	.636	.718	.816	.912
241	.067	.066	.259	.287	.431	.495	.574	.728	.814	.926
323	.067	.150	.232	.390	.442	.507	.656	.736	.834	.923
398	.083	.123	.216	.293	.390	.502	.580	.702	.820	.912
447	.050	.121	.251	.349	.417	.488	.620	.742	.802	.930
551	.033	.133	.333	.556	.467	.592	.700	.783	.833	1.000
827	.067	.117	.202	.275	.420	.550	.567	.715	.852	.945
883	.200	.133	.191	.314	.471	.514	.638	.728	.859	.948
885	.100	.150	.178	.380	.493	.552	.677	.718	.846	.952
1014	.067	.067	.167	.422	.450	.583	.713	.798	.874	1.000
1018	.111	.104	.328	.371	.411	.533	.637	.725	.797	.926
1197	.083	.083	.254	.276	.452	.504	.643	.700	.831	.918
TOTAL	.094	.113	.224	.357	.432	.518	.638	.734	.834	.939

Figure 6.16: Agreement of random summary with 3 human annotators, in precision (= recall)

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.033	.183	.316	.444	.553	.631	.718	.799	.863	.953
46	.378	.402	.412	.508	.608	.705	.774	.839	.887	.940
54	.450	.601	.641	.680	.705	.805	.830	.885	.937	.968
60	.189	.347	.309	.425	.531	.640	.775	.820	.896	.968
61	.183	.219	.330	.384	.544	.583	.751	.900	.926	.989
62	.083	.234	.360	.435	.513	.597	.680	.770	.851	.938
112	.017	.066	.240	.321	.506	.573	.680	.769	.868	.964
125	.255	.352	.415	.472	.603	.688	.742	.802	.881	.947
199	.121	.163	.264	.353	.477	.574	.681	.785	.861	.929
241	.233	.294	.249	.353	.476	.581	.673	.780	.867	.954
323	.500	.517	.534	.595	.590	.634	.696	.764	.850	.929
398	.344	.356	.343	.345	.431	.511	.589	.693	.795	.911
447	.100	.243	.457	.538	.611	.640	.714	.777	.855	.939
551	.667	.667	.667	.645	.664	.750	.980	1.000	1.000	1.000
827	.167	.133	.294	.378	.520	.611	.709	.797	.874	.957
883	.233	.333	.444	.562	.600	.643	.685	.776	.877	.949
885	.667	.667	.625	.618	.583	.619	.686	.754	.873	.970
1014	.667	.667	.833	.670	.917	.917	.800	1.000	1.000	1.000
1018	.194	.181	.328	.491	.589	.687	.718	.770	.850	.947
1197	.500	.458	.480	.516	.565	.646	.730	.811	.875	.958
TOTAL	.354	.387	.447	.483	.583	.652	.726	.818	.888	.954

Figure 6.17: Agreement of lead with 3 human annotators in precision (= recall)

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.133	.244	.261	.344	.492	.629	.730	.818	.896	.948
46	.188	.226	.317	.404	.498	.562	.647	.728	.825	.914
54	.000	.017	.203	.399	.558	.604	.677	.752	.839	.910
60	.211	.234	.293	.365	.467	.557	.694	.804	.865	.947
61	.617	.608	.500	.578	.530	.593	.745	.827	.856	.993
62	.089	.255	.338	.485	.534	.606	.700	.790	.879	.945
112	.133	.200	.350	.478	.592	.661	.771	.829	.861	.956
125	.094	.224	.362	.490	.569	.650	.710	.782	.855	.931
199	.174	.232	.357	.413	.492	.588	.700	.807	.872	.938
241	.200	.294	.372	.440	.508	.571	.678	.754	.858	.946
323	.067	.250	.309	.367	.440	.582	.711	.769	.834	.919
398	.111	.200	.290	.391	.505	.549	.637	.747	.842	.928
447	.083	.183	.349	.486	.587	.642	.727	.824	.881	.921
551	.300	.300	.183	.365	.393	.650	.787	.803	.833	1.000
827	.167	.250	.433	.529	.680	.770	.822	.823	.855	.942
883	.200	.217	.286	.387	.535	.586	.696	.790	.884	.943
885	.033	.050	.267	.459	.540	.615	.678	.728	.839	.952
1014	.267	.267	.667	.466	.667	.667	.813	1.000	1.000	1.000
1018	.233	.279	.375	.462	.539	.610	.713	.791	.859	.929
1197	.050	.130	.332	.408	.488	.610	.724	.794	.872	.958
TOTAL	.160	.231	.351	.420	.519	.611	.723	.807	.871	.952

Figure 6.18: Agreement of MEAD vs. 3 human annotators, in precision (= recall)

	Compression				
Clusters	05	10	20	30	40
2	.000	.033	.165	.244	.334
46	.183	.190	.237	.321	.428
54	.300	.299	.451	.572	.604
60	.044	.073	.171	.267	.415
61	.167	.167	.296	.355	.508
62	.000	.033	.215	.304	.415
112	.117	.161	.236	.401	.490
125	.200	.221	.316	.417	.525
199	.081	.117	.227	.337	.417
241	.217	.322	.393	.473	.546
323	.467	.283	.276	.380	.491
398	.172	.174	.297	.380	.435
447	.033	.122	.311	.404	.497
551	.467	.467	.633	.670	.678
827	.200	.217	.380	.476	.600
883	.300	.300	.335	.351	.416
885	.467	.533	.433	.441	.530
1014	.667	.667	.500	.557	.917
1018	.172	.165	.225	.324	.439
1197	.417	.342	.365	.426	.511
TOTAL	.310	.305	.358	.439	.543

Figure 6.19: Agreement of Websum with 3 human annotators in precision (= recall)

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.075	.061	.057	.122	.194	.264	.288	.384	.364	.508
46	.017	.051	.075	.121	.152	.188	.207	.245	.182	.055
54	.217	.370	.473	.511	.425	.464	.330	.338	.362	.413
60	.223	.263	.289	.363	.368	.392	.459	.356	.313	.389
61	.259	.187	.288	.443	.482	.422	.318	.333	.420	.513
62	.200	.150	.278	.290	.301	.331	.357	.356	.403	.350
112	-.014	.063	.126	.110	.226	.197	.237	.236	.287	.225
125	.038	.176	.271	.352	.429	.454	.457	.345	.279	.281
199	.065	.164	.119	.136	.195	.253	.283	.307	.285	.348
323	.098	.095	.063	.122	.039	.033	.027	.047	.071	.062
398	.061	.054	.008	.066	.213	.301	.406	.409	.439	.270
447	.182	.144	.144	.205	.259	.258	.214	.233	.353	.471
551	.224	.224	.095	.038	.046	.389	.932	1.000	1.000	1.000
827	.062	.227	.247	.281	.431	.500	.533	.406	.527	.447
883	.092	.088	.176	.180	.151	.190	.111	.114	.231	.596
885	.258	.241	.188	.212	.123	.055	.054	.113	.255	.368
1014	.238	.238	.556	.289	.667	.667	.644	1.000	1.000	1.000
1197	.217	.283	.448	.423	.435	.359	.389	.445	.517	.458
241	.128	.047	.130	.195	.261	.343	.415	.507	.568	.319
1018	.087	.135	.164	.178	.195	.266	.290	.268	.293	.251
TOTAL (20 clusters)	.127	.157	.194	.225	.274	.302	.323	.337	.354	.324

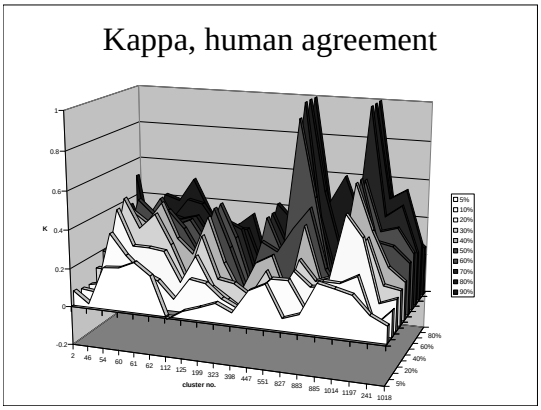


Figure 6.20: Agreement between 3 human annotators in kappa

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.090	.007	.024	.030	.088	.130	.175	.157	.184	.262
46	-.003	.051	.069	.071	.085	.094	.092	.124	.091	.006
54	.077	.188	.203	.257	.239	.288	.213	.247	.133	.196
60	.140	.146	.165	.177	.184	.158	.268	.266	.124	.195
61	.074	.086	.106	.247	.234	.199	.145	.159	.220	.270
62	.082	.047	.162	.126	.129	.130	.164	.181	.185	.192
112	.001	.018	.073	.088	.090	.098	.132	.090	.160	.076
125	.060	.092	.104	.206	.211	.224	.213	.180	.151	.101
199	.039	.107	.041	.065	.089	.086	.181	.159	.141	.169
323	.044	.046	.034	.105	.044	.017	.032	.007	.044	.026
398	.033	.037	.002	.015	.088	.121	.157	.190	.233	.132
447	.074	.075	.091	.111	.116	.112	.101	.125	.127	.254
551	.049	.108	.083	.134	.055	.227	.459	.617	.418	1.000
827	.026	.098	.114	.099	.198	.285	.216	.136	.321	.233
883	.110	.063	.065	.082	.106	.092	.070	.045	.148	.318
885	.129	.131	.056	.136	.123	.055	.069	.026	.130	.157
1014	.086	.086	.222	.182	.283	.417	.440	.600	.486	1.000
1197	.105	.120	.234	.177	.246	.167	.223	.175	.270	.187
241	.064 -	.005	.086	.070	.137	.150	.156	.274	.255	.142
1018	.060	.068	.126	.113	.079	.157	.171	.161	.099	.126
TOTAL (20 clusters)	.064	.081	.097	.116	.137	.145	.169	.171	.175	.156

Figure 6.21: Agreement of random summaries with 3 human annotators in kappa

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.030	.070	.091	.157	.218	.249	.264	.313	.314	.469
46	.174	.190	.171	.202	.239	.290	.305	.337	.290	.171
54	.315	.494	.498	.511	.461	.539	.439	.444	.471	.485
60	.195	.255	.211	.248	.260	.310	.397	.321	.320	.417
61	.167	.149	.197	.304	.375	.338	.342	.401	.410	.374
62	.117	.124	.226	.227	.239	.259	.261	.272	.298	.287
112	-.028	.018	.083	.075	.202	.178	.189	.178	.281	.314
125	.116	.204	.246	.282	.376	.413	.390	.317	.299	.291
199	.059	.103	.080	.084	.146	.189	.210	.260	.250	.302
323	.278	.262	.229	.253	.164	.139	.088	.067	.090	.062
398	.183	.151	.077	.058	.128	.170	.188	.181	.185	.125
447	.110	.152	.231	.275	.301	.282	.232	.207	.285	.326
551	.418	.418	.316	.211	.227	.408	.932	1.000	1.000	1.000
827	.080	.109	.181	.185	.296	.352	.379	.298	.377	.331
883	.129	.177	.239	.253	.218	.212	.099	.121	.200	.343
885	.444	.431	.349	.318	.191	.129	.098	.105	.242	.368
1014	.429	.429	.667	.378	.750	.750	.556	1.000	1.000	1.000
1197	.318	.323	.381	.348	.335	.302	.318	.363	.378	.428
241	.141	.109	.071	.089	.147	.201	.232	.316	.381	.315
1018	.129	.126	.163	.210	.247	.322	.299	.268	.280	.297
TOTAL (20 clusters)	.180	.198	.213	.220	.261	.284	.287	.304	.316	.300

Figure 6.22: Agreement of Lead-based with 3 human annotators

	Compression									
Clusters	05	10	20	30	40	50	60	70	80	90
2	.075	.106	.068	.087	.175	.252	.288	.366	.389	.430
46	.083	.092	.106	.123	.149	.153	.147	.160	.133	.042
54	.063	.121	.222	.305	.304	.301	.226	.215	.227	.183
60	.195	.188	.182	.212	.220	.251	.305	.291	.259	.306
61	.398	.327	.288	.430	.369	.333	.306	.288	.310	.548
62	.129	.156	.226	.273	.265	.269	.318	.335	.383	.350
112	.031	.081	.143	.166	.245	.250	.301	.284	.249	.270
125	.027	.141	.217	.292	.329	.356	.338	.274	.225	.164
199	.078	.149	.145	.140	.179	.221	.262	.322	.297	.316
323	.044	.105	.085	.087	.039	.089	.110	.073	.044	.008
398	.052	.071	.046	.085	.187	.200	.251	.280	.327	.241
447	.110	.114	.154	.224	.280	.272	.240	.280	.336	.218
551	.205	.205	-.021	-.029	-.012	.294	.606	.652	.418	1.000
827	.080	.173	.255	.282	.425	.505	.526	.338	.321	.209
883	.110	.088	.144	.137	.168	.169	.129	.162	.231	.293
885	.092	.099	.122	.205	.150	.122	.083	.052	.118	.157
1014	.200	.200	.556	.218	.500	.500	.573	1.000	1.000	1.000
1197	.094	.154	.289	.276	.273	.270	.318	.353	.393	.428
241	.141	.109	.156	.175	.221	.247	.307	.344	.415	.308
1018	.115	.118	.158	.164	.171	.215	.234	.240	.250	.147
TOTAL (20 clusters)	.109	.136	.168	.192	.230	.252	.274	.290	.290	.253

Figure 6.23: Agreement of MEAD with 3 human annotators in kappa

	Compression				
Clusters	05	10	20	30	40
2	.001	-.020	-.004	.013	.033
46	.076	.070	.056	.066	.086
54	.217	.274	.387	.457	.405
60	.098	.096	.126	.150	.188
61	.151	.086	.151	.247	.285
62	.070	.040	.147	.146	.157
112	.016	.054	.078	.127	.167
125	.071	.148	.204	.256	.311
199	.052	.110	.078	.095	.114
323	.260	.125	.063	.093	.081
398	.089	.060	.049	.079	.138
447	.074	.075	.130	.173	.203
551	.302	.302	.292	.230	.237
827	.098	.163	.232	.244	.353
883	.165	.139	.168	.112	.073
885	.333	.336	.207	.174	.130
1014	.429	.429	.444	.289	.750
1197	.284	.269	.306	.269	.294
241	.128	.133	.177	.202	.228
1018	.101	.101	.090	.087	.106
TOTAL (20 clusters)	.138	.128	.146	.159	.192

Figure 6.24: Agreement of Websum with 3 human annotators in kappa

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.53**	**0.60**	0.64	0.72	0.76	0.82	0.88	**0.93**	**0.97**
Lexical Chains	0.50	0.59	0.63						
MEAD	0.46	0.59	**0.65**	**0.73**	**0.77**	**0.84**	**0.89**	0.92	**0.97**
Random	0.34	0.46	0.56	0.63	0.68	0.76	0.83	0.89	0.96
Summarist	0.32	0.45	0.47	0.50					
Websumm	0.47	0.51	0.58	0.66	0.72	0.79	0.86	0.90	**0.97**

Figure 6.25: Cosine $(0/1)$. Average over 10 Clusters. Words and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.51**	**0.59**	0.62	0.70	0.74	0.80	**0.87**	**0.92**	0.97
Lexical Chains	0.47	0.57	0.61						
MEAD	0.42	0.55	**0.63**	**0.71**	**0.75**	**0.82**	**0.87**	0.91	**0.97**
Random	0.28	0.42	0.53	0.61	0.65	0.74	0.81	0.87	0.96
Websumm	0.45	0.50	0.57	0.65	0.70	0.77	0.84	0.89	**0.97**

Figure 6.26: Cosine $(0/1)$. Average over 10 Clusters. Words and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.54**	**0.62**	0.65	**0.73**	0.77	0.82	0.88	**0.93**	**0.97**
Lexical Chains	0.51	0.60	0.64						
MEAD	0.47	0.60	**0.66**	**0.73**	**0.78**	0.85	**0.89**	**0.93**	**0.97**
Random	0.35	0.48	0.58	0.64	0.68	0.77	0.83	0.89	**0.97**
Websumm	0.49	0.52	0.59	0.67	0.73	0.79	0.86	0.90	**0.97**

Figure 6.27: Cosine $(0/1)$. Average over 10 Clusters. Lemmas and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.52**	**0.60**	0.62	**0.71**	0.75	0.81	**0.87**	**0.92**	**0.97**
Lexical Chains	0.48	0.57	0.62						
MEAD	0.43	0.56	**0.63**	**0.71**	**0.76**	**0.82**	**0.87**	0.91	**0.97**
Random	0.29	0.43	0.54	0.61	0.65	0.74	0.81	0.87	0.96
Websumm	0.46	0.51	0.57	0.65	0.71	0.78	0.85	0.89	**0.97**

Figure 6.28: Cosine $(0/1)$. Average over 10 Clusters. Lemmas and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.55**	**0.65**	**0.70**	**0.79**	**0.84**	**0.89**	**0.94**	**0.97**	**0.99**
Lexical Chains	0.53	0.63	0.69						
MEAD	0.46	0.61	**0.70**	0.78	0.83	**0.89**	0.93	0.95	0.98
Random	0.31	0.47	0.60	0.69	0.75	0.84	0.89	0.93	0.98
Summarist	0.26	0.43	0.47	0.51					
Websumm	0.52	0.60	0.68	0.77	0.82	0.87	0.92	0.95	**0.99**

Figure 6.29: Cosine $(tf * idf)$. Average over 10 Clusters. Words and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.57**	**0.68**	**0.73**	**0.81**	**0.86**	**0.90**	**0.95**	**0.97**	**0.99**
Lexical Chains	0.56	0.66	0.71						
MEAD	0.47	0.63	0.71	0.79	0.85	**0.90**	0.93	0.95	0.98
Random	0.34	0.51	0.63	0.73	0.77	0.85	0.89	0.94	0.98
Websumm	0.56	0.64	0.71	0.79	0.84	0.89	0.93	0.95	**0.99**

Figure 6.30: Cosine $(tf * idf)$. Average over 10 Clusters. Words and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.56**	**0.66**	**0.71**	**0.80**	**0.85**	**0.89**	**0.94**	**0.97**	**0.99**
Lexical Chains	0.54	0.64	0.70						
MEAD	0.47	0.63	**0.71**	0.79	0.84	**0.90**	0.93	0.95	0.98
Random	0.32	0.48	0.62	0.71	0.76	0.84	0.89	0.93	0.98
Websumm	0.53	0.60	0.69	0.78	0.83	0.88	0.93	0.95	**0.99**

Figure 6.31: Cosine ($tf * idf$). Average over 10 Clusters. Lemmas and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.58**	**0.69**	**0.74**	**0.82**	**0.86**	**0.91**	**0.95**	**0.98**	**0.99**
Lexical Chains	0.57	0.66	0.73						
MEAD	0.49	0.64	0.72	0.80	0.85	0.90	0.93	0.95	0.98
Random	0.35	0.53	0.65	0.74	0.78	0.86	0.90	0.94	**0.99**
Websumm	0.57	0.65	0.72	0.80	0.85	0.89	0.94	0.96	**0.99**

Figure 6.32: Cosine ($tf * idf$). Average over 10 Clusters. Lemmas and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.43**	**0.47**	**0.50**	**0.59**	**0.63**	0.71	0.79	**0.88**	**0.95**
Lexical Chains	0.38	0.45	0.48						
MEAD	0.33	0.44	0.49	0.57	0.63	**0.73**	**0.81**	0.87	**0.95**
Random	0.22	0.31	0.41	0.47	0.52	0.62	0.71	0.80	0.93
Summarist	0.20	0.31	0.32	0.34					
Websumm	0.35	0.36	0.41	0.51	0.58	0.66	0.75	0.82	0.94

Figure 6.33: Unigram Overlap. Average over 10 Clusters. Words and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.42**	**0.47**	**0.48**	**0.57**	**0.61**	0.69	**0.78**	**0.86**	**0.94**
Lexical Chains	0.37	0.43	0.45						
MEAD	0.30	0.40	0.46	0.55	**0.61**	**0.70**	**0.78**	0.85	**0.95**
Random	0.18	0.29	0.38	0.45	0.50	0.59	0.68	0.78	0.93
Websumm	0.33	0.35	0.41	0.50	0.56	0.63	0.73	0.80	0.94

Figure 6.34: Unigram Overlap. Average over 10 Clusters. Words and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.56**	**0.66**	**0.71**	**0.80**	**0.85**	0.89	**0.94**	**0.97**	**0.99**
Lexical Chains	0.54	0.64	0.70						
MEAD	0.47	0.63	**0.71**	0.79	0.84	**0.90**	0.93	0.95	0.98
Random	0.32	0.48	0.62	0.71	0.76	0.84	0.89	0.93	0.98
Websumm	0.53	0.60	0.69	0.78	0.83	0.88	0.93	0.95	**0.99**

Figure 6.35: Unigram Overlap. Average over 10 Clusters. Lemmas and all POS as text representation

6.2.3 Unigram Overlap Similarity

The results obtained with these measures for all the representations chosen can be seen in Figures 6.33, 6.34, 6.35, and 6.36. Using this measure, Lead Based obtain results close to the human extracts in most of the compression rates while MEAD is ranked second.

6.2.4 Bigram Overlap Similarity

The results obtained with these measures for all the representations chosen can be seen in Figures 6.37, 6.38, 6.39, and 6.40. Using this measure, Lead Based obtain results close to the human extracts in most of the compression rates while MEAD is ranked second.

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.42**	**0.47**	**0.49**	**0.58**	**0.61**	**0.69**	**0.78**	**0.87**	0.94
Lexical Chains	0.37	0.44	0.46						
MEAD	0.31	0.41	0.47	0.56	**0.61**	**0.70**	**0.78**	0.84	**0.95**
Random	0.19	0.30	0.39	0.45	0.50	0.59	0.68	0.78	0.93
Websumm	0.34	0.36	0.41	0.50	0.56	0.64	0.73	0.80	0.94

Figure 6.36: Unigram Overlap. Average over 10 Clusters. Lemmas and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.35**	**0.38**	**0.41**	**0.51**	0.56	0.65	0.76	**0.85**	**0.94**
Lexical Chains	0.28	0.35	0.37						
MEAD	0.23	0.33	0.39	0.49	**0.57**	**0.69**	**0.78**	**0.85**	**0.94**
Random	0.12	0.20	0.29	0.36	0.43	0.54	0.65	0.76	0.91
Summarist	0.11	0.20	0.22	0.24					
Websumm	0.25	0.25	0.31	0.42	0.50	0.60	0.71	0.80	**0.94**

Figure 6.37: Bigram Overlap. Average over 10 Clusters. Words and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.35**	**0.38**	**0.40**	**0.50**	**0.54**	**0.63**	**0.74**	**0.84**	0.92
Lexical Chains	0.28	0.34	0.36						
MEAD	0.22	0.31	0.37	0.46	0.53	0.62	0.73	0.81	**0.93**
Random	0.11	0.18	0.27	0.33	0.39	0.49	0.60	0.71	0.90
Websumm	0.25	0.25	0.31	0.41	0.48	0.57	0.69	0.76	0.92

Figure 6.38: Bigram Overlap. Average over 10 Clusters. Words and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.35**	**0.38**	**0.41**	**0.51**	0.56	0.65	0.76	**0.86**	0.94
Lexical Chains	0.28	0.35	0.37						
MEAD	0.23	0.33	0.39	0.49	**0.57**	**0.69**	**0.78**	**0.85**	**0.95**
Random	0.12	0.20	0.30	0.36	0.43	0.54	0.65	0.77	0.92
Websumm	0.25	0.25	0.31	0.43	0.51	0.60	0.72	0.80	0.94

Figure 6.39: Bigram Overlap. Average over 10 Clusters. Lemmas and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.35**	**0.38**	**0.40**	**0.50**	**0.54**	**0.63**	**0.74**	**0.84**	0.92
Lexical Chains	0.28	0.34	0.36						
MEAD	0.22	0.31	0.37	0.46	0.53	**0.63**	0.73	0.81	**0.93**
Random	0.11	0.19	0.27	0.33	0.39	0.49	0.60	0.71	0.90
Websumm	0.25	0.25	0.31	0.41	0.48	0.57	0.69	0.76	**0.93**

Figure 6.40: Bigram Overlap. Average over 10 Clusters. Lemmas and nouns as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.47**	**0.55**	0.60	0.70	0.75	0.82	0.88	**0.94**	0.97
Lexical Chains	0.42	0.53	0.59						
MEAD	0.37	0.52	**0.61**	**0.70**	**0.76**	**0.84**	**0.89**	0.93	0.97
Random	0.25	0.38	0.50	0.58	0.64	0.74	0.82	0.89	0.96
Summarist	0.25	0.42	0.45	0.49					
Websumm	0.39	0.45	0.53	0.64	0.71	0.79	0.87	0.91	**0.98**

Figure 6.41: Longest Common Subsequence. Average over 10 Clusters. Words and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.48**	**0.56**	**0.61**	**0.70**	**0.75**	0.81	**0.88**	**0.93**	0.97
Lexical Chains	0.43	0.54	0.59						
MEAD	0.36	0.51	0.60	0.69	**0.75**	**0.82**	**0.88**	0.92	0.97
Random	0.24	0.38	0.50	0.58	0.64	0.73	0.80	0.88	0.96
Websumm	0.42	0.48	0.56	0.65	0.72	0.79	0.87	0.91	**0.98**

Figure 6.42: Longest Common Subsequence. Average over 10 Clusters. Words and nouns as text representation

6.2.5 Longest Common Subsequence Similarity

The results obtained with these measures for all the representations chosen can be seen in Figures 6.41, 6.42, 6.43, and 6.44. Using this measure, no system obtain better results in the majority of the cases.

6.3 Relative Utility results

We ran four experiments to compute relative utility values for a number of summarizers at ten summary lengths. We also produced relative utility values for a few baselines - lead-based and random summaries.

6.3.1 Single-document J/R values

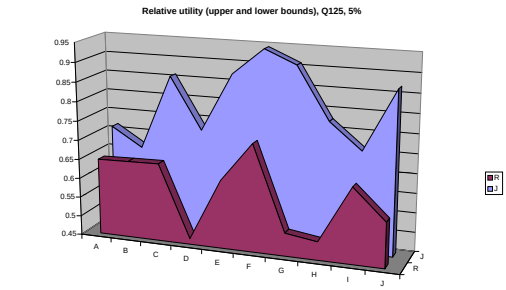
In the experiments below, J is the upper bound. R is the lower bound on the performance of an extractive summarizer. Reasonable summarizers are expected to have relative utility S in the range between R and J. Note that

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.47**	**0.56**	**0.61**	**0.70**	0.75	0.82	0.88	**0.94**	**0.98**
Lexical Chains	0.42	0.53	0.59						
MEAD	0.37	0.52	**0.61**	**0.70**	**0.76**	**0.84**	**0.89**	0.93	0.97
Random	0.26	0.39	0.51	0.59	0.65	0.74	0.82	0.89	0.97
Websumm	0.40	0.45	0.54	0.64	0.71	0.79	0.87	0.91	**0.98**

Figure 6.43: Longest Common Subsequence. Average over 10 Clusters. Lemmas and all POS as text representation

Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lead Based	**0.48**	**0.57**	**0.61**	**0.70**	**0.75**	0.81	**0.88**	**0.93**	0.97
Lexical Chains	0.43	0.54	0.60						
MEAD	0.36	0.51	0.60	0.69	**0.75**	**0.82**	**0.88**	0.92	0.97
Random	0.24	0.38	0.51	0.58	0.64	0.74	0.81	0.88	0.96
Websumm	0.42	0.48	0.56	0.66	0.72	0.80	0.87	0.91	**0.98**

Figure 6.44: Longest Common Subsequence. Average over 10 Clusters. Lemmas and nouns as text representation



	A	B	C	D	E	F	G	H	I	J
R	0.648	0.65	0.652	0.465	0.626	0.727	0.509	0.497	0.644	0.566
J	0.715	0.666	0.859	0.726	0.876	0.944	0.909	0.776	0.71	0.869

Figure 6.45: Interjudge agreement (J) and random performance (R) for cluster 125, per document, 5% target length

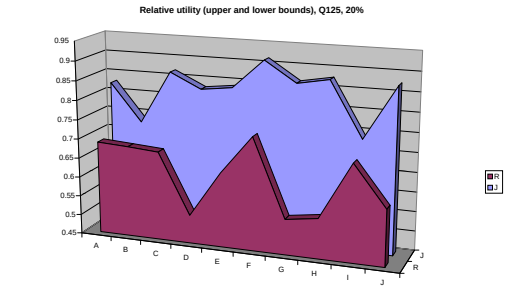
occasionally (on a particular input and at a particular summary length) a summarizer can score worse than random or better than J. However, when averaging over a number of clusters, these outliers cancel out.

Figures 6.45 and 6.46 show how single-document J and R vary by document within a cluster. The first figure is for 5% extracts and the second one – for 20% extracts. The area between the two lines is where a reasonable summarizer's performance lies.

6.3.2 Single-document RU evaluation

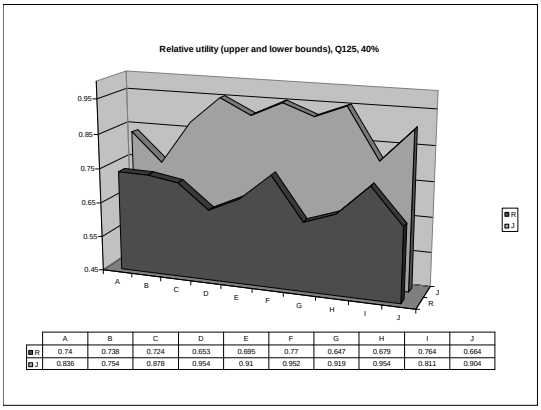
We computed J (interjudge agreement), R (random performance), S (system performance), and D (normalized system performance) over all 20 clusters (total = 200 documents). The results are presented in Figure 6.48.

We should note the concept of a random summary produced by picking random sentences given a summary length is different from the idea of R as described above. To produce R, we average over all possible $\binom{n}{e}$ combinations of e sentences out of n where the random summary method produces only *one* such combination. It should be expected, over a large sample, that RANDOM extracts perform as poorly as R and our experiments show that such is indeed the case.



	A	B	C	D	E	F	G	H	I	J
R	0.69	0.685	0.679	0.523	0.642	0.741	0.541	0.553	0.699	0.595
J	0.827	0.73	0.866	0.828	0.838	0.913	0.861	0.876	0.736	0.874

Figure 6.46: Relative utility - interjudge agreement (J) and random performance (R) for cluster 125, per document, 20% target length



	A	B	C	D	E	F	G	H	I	J
R	0.74	0.738	0.724	0.653	0.695	0.77	0.647	0.679	0.764	0.664
J	0.836	0.754	0.878	0.954	0.91	0.952	0.919	0.954	0.811	0.904

Figure 6.47: Relative utility - upper and lower bounds for cluster 125, per document, 40% target length

The single-document results tables compare MEAD with WEBSUMM and the two baselines RANDOM and LEAD.

Several interesting observations can be made looking at the data in Figure 6.48. First, random performance is quite high although certainly beatable, as shown in Figures 6.45 and 6.46. Second, both the lower bound (J) and the upper bound (R) increase with summary length. Third, even though the performances of MEAD and WEBSUMM (S) also increase with summary length, MEAD's normalized version (D) decreases slowly with summary length until the two summarizers score about the same on both S and D for longer summaries. Fourth, for summary lengths of 80% and above, R gets really close to J showing that reasonable summarization that significantly beats random at such summary lengths is quite difficult. Fifth, MEAD consistently outperforms LEAD across all summary lengths.

			MEAD		RANDOM		LEAD		WEBSUMM	
PCT	J	R	S	D	S	D	S	D	S	D
05	0.80	0.66	0.78	0.88	0.67	0.05	0.72	0.41	0.72	0.44
10	0.81	0.68	0.79	0.84	0.67	-0.02	0.73	0.42	0.73	0.44
20	0.83	0.71	0.79	0.68	0.71	0.01	0.77	0.52	0.76	0.43
30	0.85	0.74	0.81	0.64	0.75	0.10	0.80	0.55	0.79	0.44
40	0.87	0.76	0.83	0.63	0.77	0.03	0.83	0.64	0.82	0.51
50	0.89	0.79	0.85	0.61	0.79	0.01	0.86	0.63	0.85	0.55
60	0.92	0.83	0.88	0.59	0.83	0.02	0.89	0.63	0.87	0.42
70	0.94	0.86	0.91	0.58	0.87	0.08	0.92	0.69	0.90	0.48
80	0.96	0.91	0.93	0.45	0.91	0.05	0.94	0.66	0.93	0.36
90	0.98	0.96	0.97	0.37	0.96	0.04	0.98	0.68	0.97	0.53

Figure 6.48: Single-document Relative Utility

6.3.3 Multi-doc RU evaluation

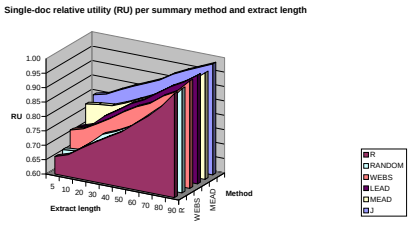
In this section, we provide multi-document RU results. Given that MEAD was the only multi-document summarizer available to us, in Figure 6.49 we only include MEAD-specific results, in addition to the two baselines: RANDOM and LEAD.

As one can see from the table, multi-document RU is slightly lower than single-document RU. We believe that this can be explained by the fact that the distribution of scores by the same judge across different articles in the same cluster is not uniform. Some documents contain only a small number of high-utility sentences and contribute to the increase in RU for single-document vs. multi-document. In addition to RU, the lower bound (R) and the upped bound (J) are also slightly lower for multi-document extracts. As a result, the normalized performance (D) is almost exactly the same in both cases. p

			MEAD		RANDOM		LEAD	
PCT	J	R	S	D	S	D	S	D
05	0.76	0.64	0.73	0.81	0.63	-0.08	0.71	0.62
10	0.78	0.66	0.75	0.76	0.65	-0.01	0.71	0.47
20	0.81	0.69	0.78	0.74	0.71	0.15	0.76	0.55
30	0.83	0.72	0.79	0.65	0.72	0.01	0.79	0.67
40	0.85	0.74	0.81	0.62	0.74	-0.06	0.82	0.72
50	0.87	0.77	0.82	0.58	0.79	0.11	0.84	0.70
60	0.88	0.80	0.84	0.52	0.81	0.00	0.86	0.66
70	0.91	0.82	0.86	0.49	0.85	0.06	0.88	0.59
80	0.92	0.84	0.88	0.45	0.89	0.03	0.90	0.55
90	0.93	0.86	0.89	0.36	0.93	-0.04	0.91	0.52

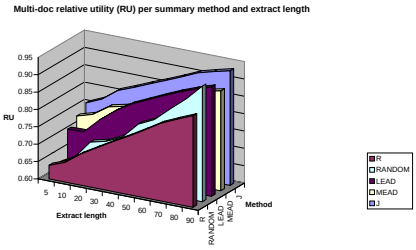
Figure 6.49: Multi-Document Relative Utility

Figures 6.50 and 6.51 summarize the results obtained through the (non-normalized) relative utility method. As the figures indicate, random performance is quite high although all non-random methods outperform it significantly. Further, in both the single- and multi-document case, MEAD outperforms LEAD for shorter summaries (5-30%). The lower bound (R) represents the average performance of all extracts at the given summary length while the upper bound (J) is the interjudge agreement among the three judges.



	5	10	20	30	40	50	60	70	80	90
R	0.66	0.68	0.71	0.74	0.76	0.79	0.83	0.86	0.91	0.96
RANDOM	0.67	0.67	0.71	0.75	0.77	0.79	0.83	0.87	0.91	0.96
WEBS	0.72	0.73	0.76	0.79	0.82	0.85	0.87	0.90	0.93	0.97
LEAD	0.72	0.73	0.77	0.80	0.83	0.86	0.89	0.92	0.94	0.98
MEAD	0.78	0.79	0.79	0.81	0.83	0.85	0.88	0.91	0.93	0.97
J	0.80	0.81	0.83	0.85	0.87	0.89	0.92	0.94	0.96	0.98

Figure 6.50: RU per summarizer and target length (Single-document)



	5	10	20	30	40	50	60	70	80	90
R	0.64	0.66	0.69	0.72	0.74	0.77	0.80	0.82	0.84	0.86
RANDOM	0.63	0.65	0.71	0.72	0.74	0.79	0.81	0.85	0.89	0.93
LEAD	0.71	0.71	0.76	0.79	0.82	0.84	0.86	0.88	0.90	0.91
MEAD	0.73	0.75	0.78	0.79	0.81	0.82	0.84	0.86	0.88	0.89
J	0.76	0.78	0.81	0.83	0.85	0.87	0.88	0.91	0.92	0.93

Figure 6.51: RU per summarizer and target length (Multi-document)

	5	10	20	30	40	50	60	70	80	90
R	0.66	0.68	0.71	0.74	0.76	0.79	0.83	0.86	0.91	0.96
Random	0.67	0.67	0.71	0.75	0.77	0.79	0.83	0.87	0.91	0.96
Websumm	0.72	0.73	0.76	0.79	0.82	0.85	0.87	0.90	0.93	0.97
Lead	0.72	0.73	0.77	0.80	0.83	0.86	0.89	0.92	0.94	0.98
MEAD	0.78	0.79	0.79	0.81	0.83	0.85	0.88	0.91	0.93	0.97
J	0.80	0.81	0.83	0.85	0.87	0.89	0.92	0.94	0.96	0.98

Figure 6.52: RU per summarizer and summary length (Single-document)

	5	10	20	30	40	50	60	70	80	90
R	0.64	0.66	0.69	0.72	0.74	0.77	0.80	0.82	0.84	0.86
Random	0.63	0.65	0.71	0.72	0.74	0.79	0.81	0.85	0.89	0.93
Lead	0.71	0.71	0.76	0.79	0.82	0.84	0.86	0.88	0.90	0.91
MEAD	0.73	0.75	0.78	0.79	0.81	0.82	0.84	0.86	0.88	0.89
J	0.76	0.78	0.81	0.83	0.85	0.87	0.88	0.91	0.92	0.93

Figure 6.53: RU per summarizer and summary length (Multi-document)

6.4 IR results

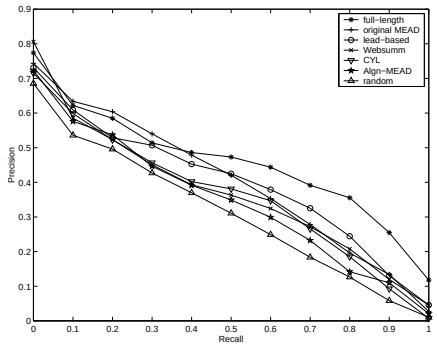


Figure 6.54: Average performance of retrieving various summaries for queries 1–20

Figure 6.54 depicts the recall-precision graphs of the average performance of retrieving various summaries for 20 queries. The summaries have 30% sentence-based length. This plot also shows the performance of retrieving the full-length documents. As shown in the plot, precision and recall tend to be inversely related as expected. We can observe that the retrieval results are generally close for different kinds of summaries as well as full-length documents for the recall region less than 0.1. In the recall region between 0.1 and 0.4, MEAD summary performs better than other kinds of summaries and slightly better than full-length documents. For the recall region greater than 0.5, the retrieval result of full-length documents achieves a higher precision than summaries. In this region, the lead-based summary performs slightly better among other summaries albeit small difference. Finally, the random summary consistently obtains inferior performance over the whole recall spectrum.

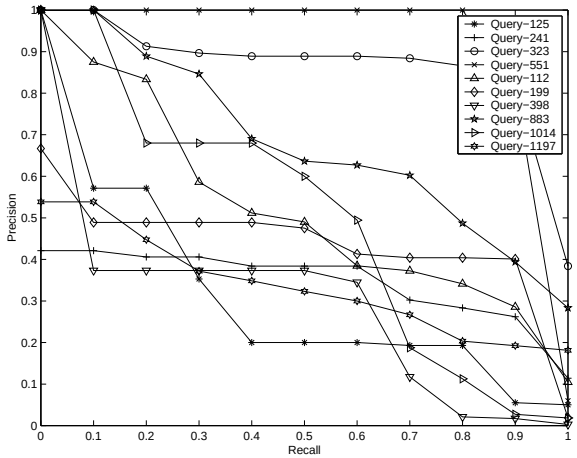


Figure 6.55: Mono-lingual retrieval of English full-length documents for queries 1 – 10

Figures 6.55 and 6.56 depict the recall-precision graphs of mono-lingual retrieval of English full-length documents for the first 10 and the next 10 queries respectively. As for typical behavior, precision and recall tend to be inversely related.

Figures 6.57 and 6.58 depict the recall-precision graphs of mono-lingual retrieval of Chinese full-length documents for the first 10 and the next 10 queries respectively. In general, Chinese mono-lingual retrieval is slightly less effective than that of English mono-lingual retrieval.

Figures 6.59 and 6.60 depict the recall-precision graphs of cross-lingual retrieval of Chinese full-length documents using English queries for the first 10 and the next 10 queries respectively. Both English cross-lingual retrieval and Chinese mono-lingual retrieval involve retrieving Chinese documents. In general, English cross-lingual

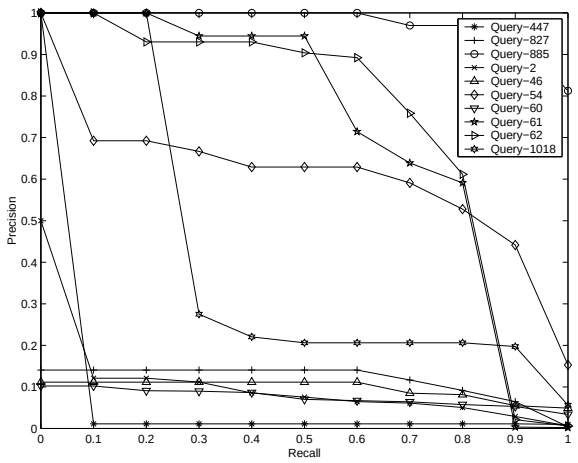


Figure 6.56: Mono-lingual retrieval of English full-length documents for queries 11–20

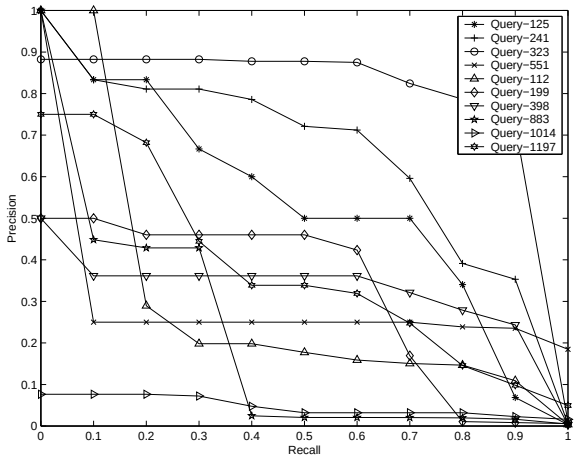


Figure 6.57: Mono-lingual retrieval of Chinese full-length documents for queries 1–10

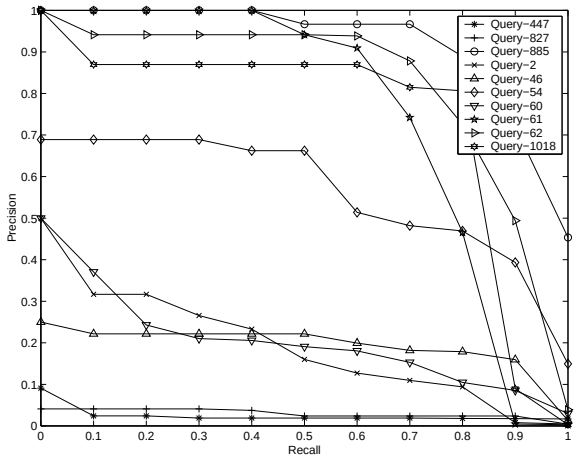


Figure 6.58: Mono-lingual retrieval of Chinese full-length documents for queries 11–20

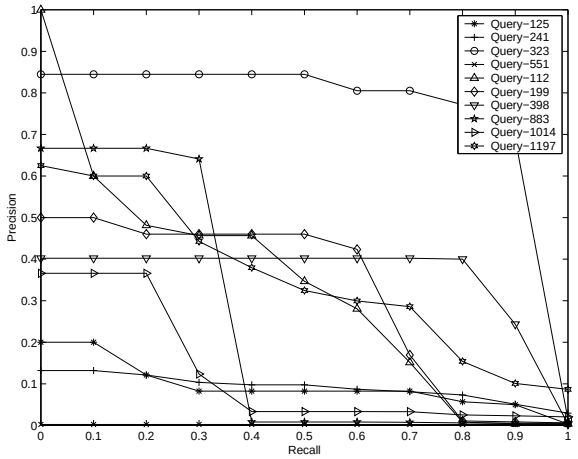


Figure 6.59: Cross-lingual retrieval of Chinese full-length documents for queries 1–10

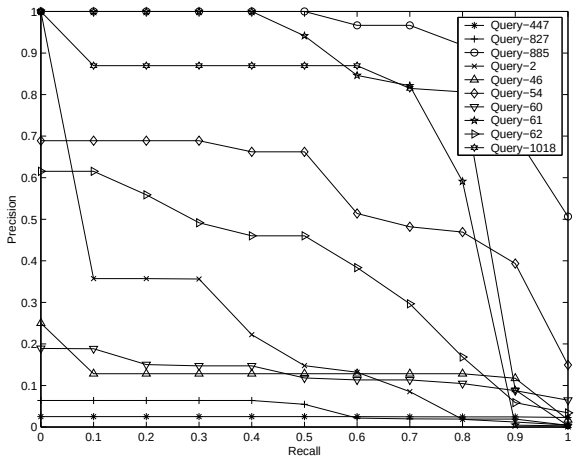


Figure 6.60: Cross-lingual retrieval of Chinese full-length documents for queries 11–20

retrieval is less effective than that of Chinese mono-lingual retrieval. Nevertheless, the average performance is quite satisfactory as compared with the recent cross-lingual track in TREC-9 evaluation.

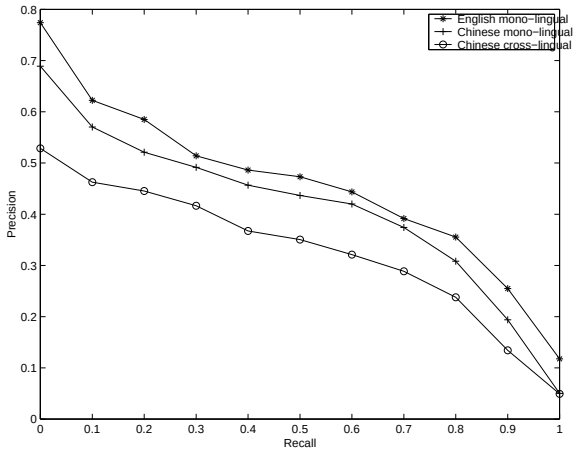
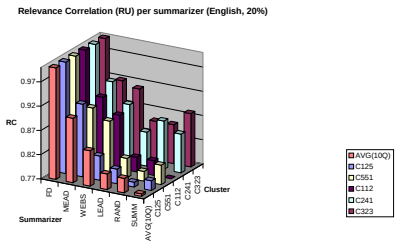


Figure 6.61: Average performance of retrieving full-length documents for queries 1–20

Figures 6.61 depicts the recall-precision graphs of the average performance of retrieving full-length documents for English mono-lingual, Chinese mono-lingual, and English cross-lingual retrieval.

6.5 Relevance correlation results

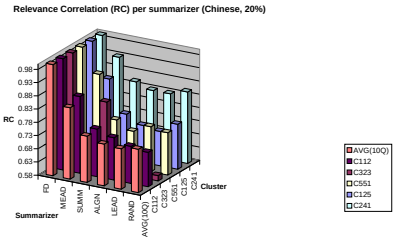
We present several results using Relevance Correlation. Figures 6.62 and 6.63 show how RC changes depending on the summarizer and the language used. In these figures, an RC value of 1 is obtained when full documents (FD) are compared to themselves. All surrogates for the set of full documents get lower scores. One can notice that even random extracts get a relatively high RC score. It is also worth observing that Chinese summaries score lower than their corresponding English summaries.



	FD	MEAD	WEBS	LEAD	RAND	SUMM
AVG(10Q)	1	0.903	0.843	0.802	0.8	0.775
C125	1	0.92	0.82	0.8	0.78	0.79
C551	1	0.9	0.88	0.81	0.79	0.81
C112	1	0.91	0.88	0.8	0.8	0.77
C241	1	0.93	0.89	0.84	0.87	0.85
C323	1	0.92	0.91	0.85	0.85	0.88

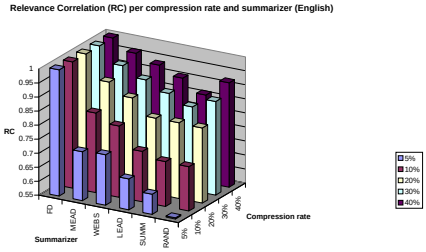
Figure 6.62: Relevance correlation per summarizer (English 20%)

Figure 6.64 shows the effects of summary length and summarizers on RC.



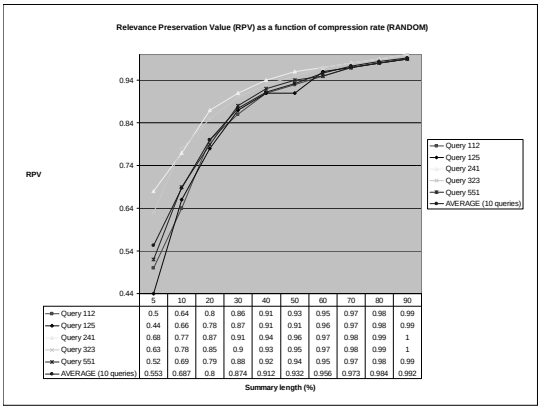
	FD	MEAD	SUMM	ALGN	LEAD	RAND
AVG(10Q)	1	0.85	0.755	0.738	0.733	0.744
C112	1	0.87	0.76	0.74	0.72	0.71
C323	1	0.66	0.84	0.59	0.58	0.6
C551	1	0.91	0.75	0.72	0.75	0.74
C125	1	0.87	0.75	0.72	0.71	0.75
C241	1	0.93	0.85	0.83	0.83	0.85

Figure 6.63: Relevance correlation per summarizer (Chinese, 20%)



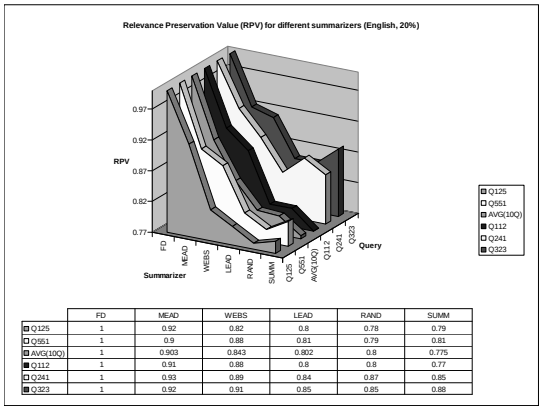
	FD	MEAD	WEBS	LEAD	SUMM	RAND
5%	1	0.724	0.73	0.66	0.622	0.554
10%	1	0.834	0.804	0.73	0.71	0.708
20%	1	0.916	0.876	0.82	0.82	0.818
30%	1	0.946	0.912	0.88	0.848	0.884
40%	1	0.962	0.936	0.906	0.862	0.922

Figure 6.64: Relevance correlation per summary length and summarizer



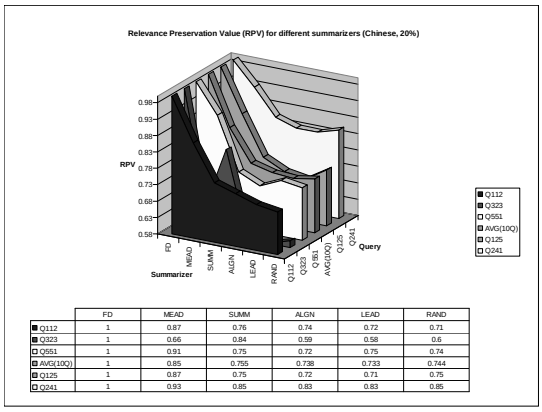
	5	10	20	30	40	50	60	70	80	90
Query 112	0.5	0.64	0.8	0.86	0.91	0.93	0.95	0.97	0.98	0.99
Query 125	0.44	0.66	0.78	0.87	0.91	0.91	0.96	0.97	0.98	0.99
Query 241	0.68	0.77	0.87	0.91	0.94	0.96	0.97	0.98	0.99	1
Query 323	0.63	0.78	0.85	0.9	0.93	0.95	0.97	0.98	0.99	1
Query 551	0.52	0.69	0.79	0.88	0.92	0.94	0.95	0.97	0.98	0.99
AVERAGE (10 queries)	0.553	0.687	0.8	0.874	0.912	0.932	0.956	0.973	0.984	0.992

Figure 6.65: Relevance correlation as a function of compression rate (RANDOM)



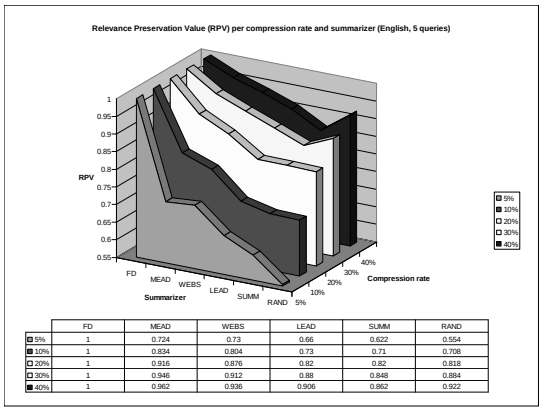
	FD	MEAD	WEBS	LEAD	RAND	SUMM
Q125	1	0.92	0.82	0.8	0.78	0.79
Q551	1	0.9	0.88	0.81	0.79	0.81
AVG(10Q)	1	0.903	0.843	0.802	0.8	0.775
Q112	1	0.91	0.88	0.8	0.8	0.77
Q241	1	0.93	0.89	0.84	0.87	0.85
Q323	1	0.92	0.91	0.85	0.85	0.88

Figure 6.66: Relevance correlation for different summarizers (English, 20%)



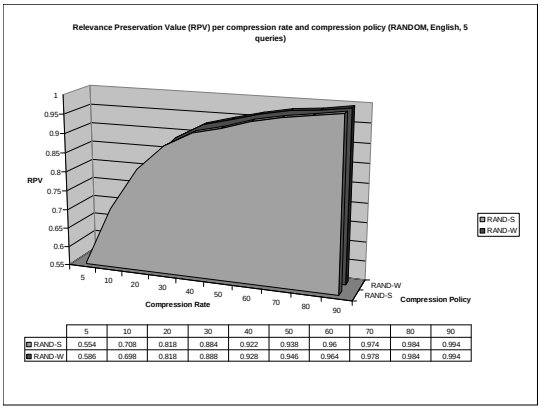
	FD	MEAD	SUMM	ALGN	LEAD	RAND
Q112	1	0.87	0.76	0.74	0.72	0.71
Q323	1	0.66	0.84	0.59	0.58	0.6
Q551	1	0.91	0.75	0.72	0.75	0.74
AVG(10Q)	1	0.85	0.755	0.738	0.733	0.744
Q125	1	0.87	0.75	0.72	0.71	0.75
Q241	1	0.93	0.85	0.83	0.83	0.85

Figure 6.67: Relevance correlation for different summarizers (Chinese, 20%)



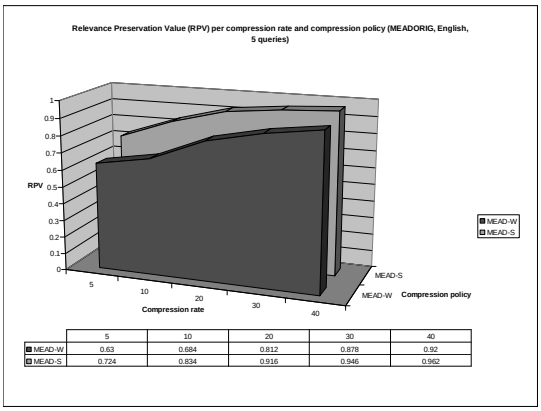
	FD	MEAD	WEBS	LEAD	SUMM	RAND
5%	1	0.724	0.73	0.66	0.622	0.554
10%	1	0.834	0.804	0.73	0.71	0.708
20%	1	0.916	0.876	0.82	0.82	0.818
30%	1	0.946	0.912	0.88	0.848	0.884
40%	1	0.962	0.936	0.906	0.862	0.922

Figure 6.68: Relevance correlation per compression rate and summarizer (English, 5 queries)



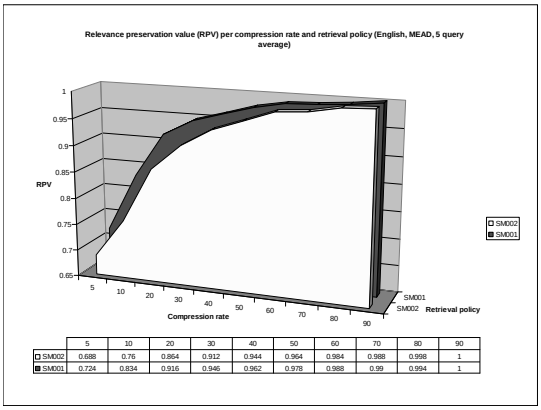
	5	10	20	30	40	50	60	70	80	90
RAND-S	0.554	0.708	0.818	0.884	0.922	0.938	0.96	0.974	0.984	0.994
RAND-W	0.586	0.698	0.818	0.888	0.928	0.946	0.964	0.978	0.984	0.994

Figure 6.69: Relevance correlation per compression rate and compression policy (RANDOM, English, 5 queries)



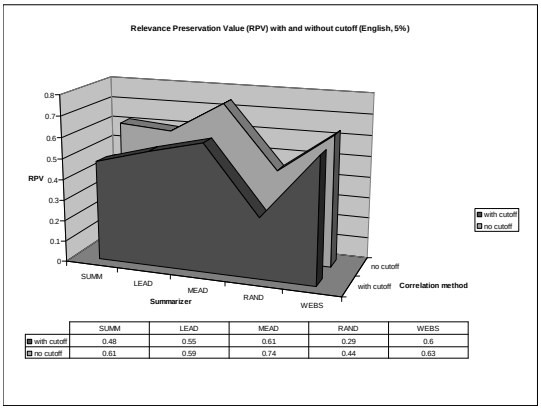
	5	10	20	30	40
MEAD-W	0.63	0.684	0.812	0.878	0.92
MEAD-S	0.724	0.834	0.916	0.946	0.962

Figure 6.70: Relevance correlation per compression rate and compression policy (MEADORIG, English, 5 queries)



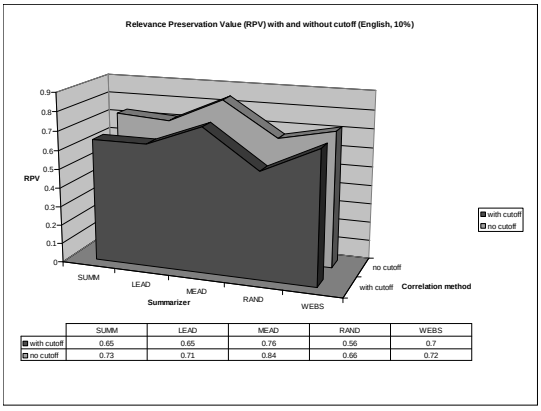
	5	10	20	30	40	50	60	70	80	90
SM002	0.688	0.76	0.864	0.912	0.944	0.964	0.984	0.988	0.998	1
SM001	0.724	0.834	0.916	0.946	0.962	0.978	0.988	0.99	0.994	1

Figure 6.71: Relevance correlation per compression rate and compression policy (MEADORIG, English, 5 query average)



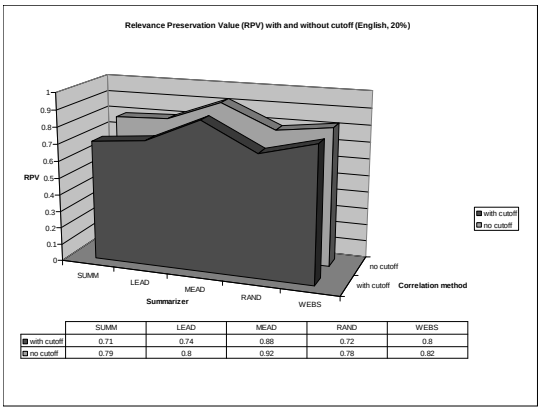
	SUMM	LEAD	MEAD	RAND	WEBS
with cutoff	0.48	0.55	0.61	0.29	0.6
no cutoff	0.61	0.59	0.74	0.44	0.63

Figure 6.72: Relevance correlation with and without cutoff (English, 5%)



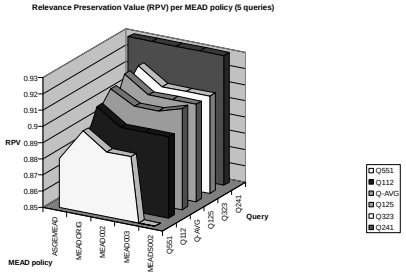
	SUMM	LEAD	MEAD	RAND	WEBS
with cutoff	0.65	0.65	0.76	0.56	0.7
no cutoff	0.73	0.71	0.84	0.66	0.72

Figure 6.73: Relevance correlation with and without cutoff (English, 10%)



	SUMM	LEAD	MEAD	RAND	WEBS
with cutoff	0.71	0.74	0.88	0.72	0.8
no cutoff	0.79	0.8	0.92	0.78	0.82

Figure 6.74: Relevance correlation with and without cutoff (English, 20%)



	ASGEMEAD	MEADORIG	MEAD002	MEAD003	MEADS002
Q551	0.88	0.9	0.89	0.89	
Q112	0.86	0.91	0.9	0.9	0.9
Q-AVG	0.886	0.916	0.908	0.908	0.9125
Q125	0.87	0.92	0.91	0.91	0.91
Q323	0.89	0.92	0.91	0.91	0.91
Q241	0.93	0.93	0.93	0.93	0.93

Figure 6.75: Relevance correlation for different MEAD parameters

Chapter 7

Conclusion

We presented what we believe is the largest collaborative effort ever to build an annotated corpus for text summarization along with a battery of methods for producing extractive summarizers and running comparisons of multiple summarizers, both single-document and multi-document.

We made several interesting contributions to text summarization:

First, we observed that different metrics rank summaries differently, although most of them showed that "intelligent" summarizers outperform lead-based summaries which is encouraging given that previous results had cast doubt on the ability of summarizers to do better than simple baselines.

Second, we introduced a new evaluation metric, Relevance Correlation, which can be used to perform large-scale summary evaluations over large corpora.

Third, we also performed a number of experiments which will be described in detail elsewhere - namely, comparison of manual extracts and manual summaries, cross-lingual summarization using sentence alignment, and evaluation of cross-lingual summaries.

Fourth, we developed a summarization toolkit including a modular state-of-the art summarizer: single-document, multi-document, generic, query-based.

Fifth, we developed a summarization evaluation toolkit allowing comparisons between extractive and non-extractive summaries.

Sixth, we performed the first ever large-scale evalatuation of summarization using Relative Utility and Relevance Correlation, comparing them to more established evaluation measures.

Seventh, we confirmed that the different metrics have different properties wrt. scalability, applicability to multi-document summaries, ability to include human agreement, etc. Figure 7.1 is a meta evaluation of all evaluation metrics that we used.

Property	Prec, recall	Kappa	Normalized RU	Word overlap, cosine, LCS	Relevance Correlation
Agreement between human extracts	X	X	X	X	X
Agreement human extracts and automatic extracts	X	X	X	X	X
Agreement human summaries/extracts				X	
Non-binary decisions			X	X	
Takes random agreement into account by design		X	X		
Full documents vs. extracts				X	X
Systems with different sentence segmentation				X	X
Multi-document extracts	X	X	X	X	
Full corpus coverage				X	X

Figure 7.1: Properties of evaluation metrics used in this project

Finally, we produced what we believe is the largest and most complete annotated corpus for further research in text summarization.

7.1 Main contributions

There are four main research areas in text summarization. Here are our contributions to each of them:

- Relevance correlation: compared to established evaluation metrics

- Relative utility: large-scale evaluation
- Comparison of query-based and generic summarization
- Comparison of manual extracts and manual summaries
- Cross-lingual summarization using alignment
- Evaluation of cross-lingual summaries

7.2 Technical accomplishments

- Develop a summarization toolkit including a modular state-of-the-art summarizer: single-document, multi-document, generic, and query-based
- Develop a summarization evaluation toolkit allowing comparisons between extractive and non-extractive summaries
- Produce a very large scale annotated corpus for further research in text summarization

7.3 Future work

- Analysis of human data for subsumption
- Evaluate trainable framework
- Fact-based evaluation
- Task-based evaluation
- Determine optimal compression rate

We will investigate the connection between RU, subsumption and the taxonomy of cross-document relationships (such as paraphrase, follow-up, elaboration, etc.) set forth in Cross-Document Structure Theory (CST) (Radev, 2000; Zhang et al., 2002).

Perhaps the most interesting and challenging aspect of multi-document summarization is related to the fact that techniques are needed to effectively compute specific relations that hold across different sources.

One of such relationships is, for example, cross document co-reference, where the system needs to identify if linguistic expressions from different sources refer to the same entity in the world. For example, the expression (1) “British Prime Minister”, and (2) “Tony Blair”, in two different documents can refer to the same entity in the world given that (1) refers to the Prime Minister of Britain in the year 2001. This is not trivial, because the expression “British Prime Minister” could refer to different entities at different times. Not only entity co-reference is needed but also event co-reference, where the system should be able to establish, for example, that “today’s tragic events” and “the Sept. 11 terrorist attacks” refer indeed to the same terrorist attacks on the Twin Towers.

As MDS systems need to address the problem of identifying redundancy and differences across documents, it is useful to take a look at some preliminary characterizations of these two notions. Following Mani (2001), redundancy across document can be characterized using the following criteria:

- two text elements are semantically equivalent when they have the same meaning. This is the linguistic notion of paraphrase;
- two text elements are string-identical when they are the same string (string identical does not entail semantic equivalence, remember the “Prime Minister” example);
- two text elements are informationally equivalent if they contain the same information: one can be used instead of the other without or with minimum lose of information;

- one text element A subsumes a text element B if the information in B is contained in A (A contains additional information);

Differences can be characterized in terms of informational equivalence and information subsumption. One piece of information in one document that cover a piece of information in another can be seen as different in the level of detail or perspective. Differences across documents is still an open research topic. Radev (2000) has identified a set of 24 relations across documents, some related to the problem of detecting differences (e.g., elaboration, contradiction, refinement, etc.).

Finally, we need to mention that the use of relative utility is not limited to the evaluation of sentence extracts. We will investigate its applicability to other evaluation tasks, such as ad-hoc retrieval and word sense disambiguation.

7.4 Acknowledgments

We thank Frederick Jelinek, Sanjeev Khudanpur, and the staff of the Center for Language and Speech Processing, Johns Hopkins University for their hospitality. The 2001 Summer Workshop at Johns Hopkins University was sponsored by the National Science Foundation via Grant No. IIS-0097467, which included support from the Defense Advanced Research Projects Agency.

We would also like to acknowledge the following individuals and organizations:

- Our Johns Hopkins hosts: Jacob Laderman, Laura Graham, Bill Byrne,
- Inderjeet Mani, Chin-Yew Lin, and Greg Silber for providing us with summarizers that we used in our experiments,
- Breck Baldwin, David Day, Sasha Blair-Goldensohn, Dan Melamed, Hamish Cunningham, Ralph Weischedel, Sean Boisen, for providing us with useful software and/or ideas,
- The LDC folks who helped annotate a large amount of data: Stephanie Strassel, Chris Cieri, David Graff,
- Donna Harman and Paul Over at NIST who gave us access to the DUC corpus.
- Mike Topper and Adam Winkel at the University of Michigan who wrote additional MEAD-related code after the end of the workshop. Mike also wrote parts of Appendix A.

Finally, we want to thank the government sponsors and the leadership at the Johns Hopkins Center for Language and Speech Processing for realizing the importance of regularly facilitating collaborative research efforts like the JHU workshop.

Appendix A

User documentation

A.1 Introduction

A.1.1 What is automatic text summarization

According to Mani (2001), ”the goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user’s or application’s need”.

A.1.2 Sentence extraction

Extractive summarization is the most robust method for text summarization. It involves assigning *salience* scores to some units (e.g., sentences or paragraphs) of a document or a set of documents and extracting these with the highest scores.

A.1.3 MEAD

MEAD is a publicly available toolkit for multi-lingual summarization and evaluation. The toolkit implements multiple summarization algorithms (at arbitrary compression rates) such as position-based, TF*IDF, largest common subsequence, and keywords. The methods for evaluating the quality of the summaries are both intrinsic (such as percent agreement, precision/recall, and relative utility) and extrinsic (document rank).

MEAD is written in Perl and requires a number of external packages to run. A full list of such packages is included in the Downloading and Installation Sections.

The current release, 3.04beta, includes support for English and Mandarin in a Solaris environment. We believe that porting to Linux is fairly straightforward. Please contact the mailing list (see below) if you are interested in porting MEAD to Linux.

Adding new (human) languages should be equally easy. Please contact the mailing list if you are interested.

A.1.4 MEAD functionality

MEAD can perform many different summarization tasks.

- Extractive single-document summarization
- Extractive multi-document summarization
- Baseline summarization
  - Lead-based
  - Random
- Monolingual summarization in different languages

  - English
  - Chinese

- Query-based summarization
- Evaluation

  The MEAD evaluation toolkit allows several ways of performing comparisons.

  - human-human agreement
  - computer-human agreement
  - computer-computer agreement

  Four evaluation methods are currently available:

  - Co-selection: precision/recall, Kappa
  - Content-based
  - Relative utility
  - Relevance correlation

A.1.5 Sample scenarios

MEAD can be used by many types of users. Here are a few scenarios in which MEAD can come in handy.

- Evaluate an existing summarizer
- Build a summarizer from scratch
- Test a summarization feature
- Test a new evaluation metric
- Test a short-query machine translation system

A.2 Downloading

A.2.1 Internal software

Internal software is the software that is directly developed by the MEAD team. All internal software can be downloaded from the jhu website. the url is:

http://www.clsp.jhu.edu/ws2001/groups/asmd/

To get started, only MEAD306.tar.gz is needed.

- MEAD306.tar.gz
  - The MEAD summarizer itself.
- LEAD & RANDOM Extractor
  - Lead-based and Random Summarizers - Included in the MEAD306 distribution.

A.2.2 External software

External software is the software that is used with MEAD, but was not developed by the MEAD team. You will need expat, XML::Parser, XML::Writer, and a few other modules (see below for a full list) to get started.

We have included the essential packages of external software with the MEAD distribution, but if you wish to download them and other useful packages yourself, we include these and other packages here.

- **Perl 5.5 or above**
  - http://www.perl.com
- **expat – needed**
  - http://sourceforge.net/projects/expat/
- **XML::Parser - needed**
  - http://www.cpan.org/authors/id/C/CO/COOPERCL/XML-Parser.2.30.tar.gz
- **XML::Writer - needed**
  - http://www.cpan.org/authors/id/DMEGG/XML-Writer-0.4.tar.gz
- **XML::Treebuilder - optional**
  - http://www.cpan.org/authors/id/S/SB/SBURKE/XML-TreeBuilder-3.08.tar.gz
- **HTML::TagSet - optional**
  - http://www.cpan.org/authors/id/S/SB/SBURKE/HTML-Tagset-3.03.tar.gz
- **HTML::Tree - optional**
  - http://www.cpan.org/authors/id/S/SB/SBURKE/HTML-Tree-3.11.tar.gz
- **HTML::Element - optional**
  - included in HTML::Tree
- **Support Vector Machines (SVM) : for trainable summarization only**
  - http://ais.gmd.de/~thorsten/svm_light/
- **SMART: for evaluation by Relevance Correlation only**
  - ftp://ftp.cs.cornell.edu/pub/smart/
- **LT-XML – optional**
  - http://www.ltg.ed.ac.uk/software/sgml/index.html

A.3 Architecture

A.3.1 Conceptual Directories

- MEAD_TEST = directory to store test files.
- MEAD_DIR = base directory to install MEAD under.
  - PROGRAM_DIR = $MEAD_DIR/programs
    - SCRIPT_DIR = $PROGRAM_DIR/scripts
  - LIBRARY_DIR = $MEAD_DIR/libraries
  - DATA_DIR = $MEAD_DIR/doc
  - DATA_DIR = $MEAD_DIR/data
    - COLLECTIONS_DIR = $DATA_DIR/collections
  - DTD_DIR = $MEAD_DIR/dtd
  - EXT_DIR - $MEAD_DIR/extensions

A.3.2 Main Objects

The DTDs describing the XML objects used in MEAD are listed at the end of this document.

Cluster

A cluster object lists the names of the documents that will be summarized.

```
 <?xml version='1.0'?>
<CLUSTER LANG="">
        <D DID="D-19980902_007.e" />
        <D DID="D-19980831_007.e" />
        <D DID="D-19980819_012.e" />
        <D DID="D-19981021_011.e" />
        <D DID="D-19980923_017.e" />
        <D DID="D-19981105_011.e" />
        <D DID="D-19981013_007.e" />
        <D DID="D-19980731_003.e" />
        <D DID="D-19980804_012.e" />
        <D DID="D-19980903_004.e" />
</CLUSTER>
```

Figure A.1: Cluster object

Docjudge

A docjudge object describes the retrieval ranking obtained from the search engine (Smart) given a query.

```
<?xml version='1.0'?>
<!DOCTYPE DOC-JUDGE SYSTEM "/export/ws01summ/dtd/docjudge.dtd">
<DOC-JUDGE QID="Q-2-E" SYSTEM="SMART" LANG="ENG">
    <D DID="D-19981007_018.e" RANK="1" SCORE="9.0000"
CORR-DOC="D-19981007_023.c"/>
    <D DID="D-19980925_013.e" RANK="2" SCORE="8.0000"
CORR-DOC="D-19980925_015.c"/>
    <D DID="D-20000308_013.e" RANK="3" SCORE="7.0000"
CORR-DOC="D-20000308_016.c"/>
    <D DID="D-19990517_005.e" RANK="4" SCORE="6.0000"
CORR-DOC="D-19990517_004.c"/>
    <D DID="D-19981017_015.e" RANK="4" SCORE="6.0000"
CORR-DOC="D-19981017_008.c"/>
    <D DID="D-19990107_019.e" RANK="12" SCORE="5.0000"
CORR-DOC="D-19990107_021.c"/>
    <D DID="D-19990713_010.e" RANK="12" SCORE="5.0000"
CORR-DOC="D-19990713_011.c"/>
    <D DID="D-19991207_006.e" RANK="12" SCORE="5.0000"
CORR-DOC="D-19991207_007.c"/>
    <D DID="D-19990913_001.e" RANK="20" SCORE="4.0000"
CORR-DOC="D-19990913_003.c"/>
    <D DID="D-19980609_005.e" RANK="20" SCORE="4.0000"
CORR-DOC="D-19980609_004.c"/>
    <D DID="D-19990825_018.e" RANK="1962" SCORE="0.0000"
CORR-DOC="D-19990825_018.c"/>
    <D DID="D-19990924_047.e" RANK="1962" SCORE="0.0000"
CORR-DOC="D-19990924_050.c"/>
</DOC-JUDGE>
```

Figure A.2: Docjudge object

Docpos

A docpos object is a document with Part of Speech Tags.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE DOCPOS SYSTEM "../../../../../dtd/docpos.dtd" >
<DOCPOS DID='D-19970701_001.e' DOCNO='1' LANG='ENG'
CORR-DOC='D-19970701_001.c'>
<BODY>
<HEADLINE>
<S PAR='1' RSNT='1' SNO='1'> <W C='JJ'>Solemn</W> <W
C='NN'>ceremony</W> <W C='VBZ'>marks</W> <W
C='NNP'>Handover</W> </S>
</HEADLINE>
<TEXT>
<S PAR='2' RSNT='1' SNO='2'><W C='DT'>A</W> <W
C='JJ'>solemn</W><W C=',',>,</W> <W C='JJ'>historic</W> <W
C='NN'>ceremony</W> <W C='VBZ'>has</W> <W C='VBN'>marked</W> <W
C='DT'>the</W> <W C='NN'>resumption</W>
<W C='IN'>of</W> <W C='DT'>the</W> <W C='NN'>exercise</W> <W
C='IN'>of</W> <W C='NN'>sovereignty</W> <W
C='IN'>over</W> <W C='NNP'>Hong</W> <W C='NNP'>Kong</W> <W
C='IN'>by</W> <W C='DT'>the</W> <W
C='NNS'>People</W><W C='POS'>'s</W> <W C='NNP'>Republic</W> <W
C='IN'>of</W> <W C='NNP'>China</W><W
C='.'>.</W></S>
<S PAR='3' RSNT='1' SNO='3'><W C='PRP$'>His</W> <W
C='NNP'>Royal</W> <W C='NNP'>Highness</W> <W
C='NNP'>The</W> <W C='NNP'>Prince</W> <W C='IN'>of</W> <W
C='NNP'>Wales</W> <W C='CC'>and</W> <W
C='DT'>the</W> <W C='NNP'>President</W> <W C='IN'>of</W> <W
C='DT'>the</W> <W C='NNS'>People</W><W
C='POS'>'s</W> <W C='NNP'>Republic</W> <W C='IN'>of</W> <W
C='NNP'>China</W> <W C='('>(</W><W
C='NNP'>PRC</W><W C=')'>)</W> <W C='NNP'>HE</W> <W C='NNP'>Mr</W>
<W C='NNP'>Jiang</W> <W
C='NNP'>Zemin</W> <W C='DT'>both</W> <W C='NN'>spoke</W> <W
C='IN'>at</W> <W C='DT'>the</W> <W
C='NN'>ceremony</W><W C=',',>,</W> <W C='WDT'>which</W> <W
C='VBD'>straddled</W> <W C='NN'>midnight</W> <W
C='IN'>of</W> <W C='NNP'>June</W> <W C='CD'>30</W> <W
C='CC'>and</W> <W C='NNP'>July</W> <W
C='CD'>1</W><W C='.'>.</W></S>
<S PAR='4' RSNT='1' SNO='4'><W C='DT'>The</W> <W
C='NN'>ceremony</W> <W C='VBD'>was</W> <W
C='VBN'>telecast</W> <W C='JJ'>live</W> <W C='IN'>around</W> <W
C='DT'>the</W> <W C='NN'>world</W><W
C='.'>.</W></S>
</TEXT>
</BODY>
</DOCPOS>
```


Figure A.3: Docpos object

Document

A document contains the text that is going to be summarized

```
<?xml version='1.0'?>
<!DOCTYPE DOCUMENT SYSTEM '/afs-clair/MEAD3/dtd/document.dtd'>
<DOCUMENT DID='D-19970701_001.e' DOCNO ='1' LANG='ENG' >
<EXTRACTION-INFO SYSTEM="./hkmead.pl Centroid 1 Position 1
Length 9" RUN="" COMP
RESSION="20" QID="D-19970701_001.e"/><BODY>
<TEXT>

The ceremony took place in the Grand Hall of the Hong Kong Convention
and Exhibition Centre (HKCEC) Extension and was attended by some 4,000
guests, including foreign ministers and dignitaries from more than 40
countries and international organisations, and about 400 of the
world's media. Representing China were Mr Jiang; HE Mr Li Peng,
Premier of the State Council of the PRC; HE Mr Qian Qichen, Vice
Premier of the State Council of the PRC; General Zhang Wannian, Vice
Chairman of the Central Military Commission of the PRC; and HE Mr Tung
Chee Hwa, the Chief Executive of the Hong Kong Special Administrative
Region (HKSAR) of the PRC.  This was followed at the stroke of
midnight by the playing of the Chinese National Anthem and the raising
of the Chinese national flag and the flag of the Hong Kong Special
Administrative Region (HKSAR) within the first minute of the new day
(Tuesday).  Entry of Guards of Honour Entry of Officiating Parties
Salute by Guards of Honour Speech by His Royal Highness The Prince of
Wales Entry of Flag Parties British National Anthem Lowering of Union
and Hong Kong Flags

Chinese National Anthem Raising of Chinese and Hong Kong Special
Administrative Region Flags Departure of Flag Parties Speech by
President of the People's Republic of China, Mr Jiang Zemin Departure
of Officiating Parties

Departure of Guards of Honour
</TEXT>
</BODY>
</DOCUMENT>
```

Figure A.4: Document object

Extract

An Extract contains a list of sentences that will be used in the summary. Sentences are sorted in the order they appear.

Query

A query object describes the text of a retrieval query (in English or Chinese).

Sentalign

A Sentalign object describes the sentence mappings between two translations of the same document.

Sentjudge

A sentjudge object is used to describe sentence utility scores given by judges to individual sentences in a document or cluster.

Summary

The Summary is the final output from the summarization process.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EXTRACT SYSTEM "/afs-clair/MEAD3/dtd/extract.dtd">

<EXTRACT QID="Group_551" COMPRESSION="20"
SYSTEM="./hkmead.pl Centroid 1 Position 1 Length 9" LANG="ENG">
<S ORDER="1" DID="D-19980731_003.e" SNO="2" />
<S ORDER="2" DID="D-19980731_003.e" SNO="3" />
<S ORDER="3" DID="D-19980804_012.e" SNO="2" />
<S ORDER="4" DID="D-19980819_012.e" SNO="2" />
<S ORDER="5" DID="D-19980819_012.e" SNO="5" />
<S ORDER="6" DID="D-19980831_007.e" SNO="5" />
<S ORDER="7" DID="D-19980902_007.e" SNO="2" />
<S ORDER="8" DID="D-19980903_004.e" SNO="2" />
<S ORDER="9" DID="D-19980903_004.e" SNO="3" />
<S ORDER="10" DID="D-19980923_017.e" SNO="4" />
<S ORDER="11" DID="D-19981021_011.e" SNO="3" />
<S ORDER="12" DID="D-19981021_011.e" SNO="4" />
<S ORDER="13" DID="D-19981105_011.e" SNO="2" />
<S ORDER="14" DID="D-19981105_011.e" SNO="3" />
<S ORDER="15" DID="D-19981105_011.e" SNO="7" />
</EXTRACT>
```

Figure A.5: Extract object

```
<?xml version='1.0'?>
<!DOCTYPE QUERY SYSTEM "../../../dtd/query.dtd" >
<QUERY QID="Q-551-E" QNO="551" TRANSLATED="NO">
<TITLE>
Natural disaster victims aided
</TITLE>
</QUERY>
```

Figure A.6: Query object

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SENTALIGN SYSTEM "/export/ws01summ/dtd/sentalign.dtd">
<SENTALIGN ENG="20000119_002.e" CHI="20000119_002.c" LANG="english-chinese">
<SENT ORDER="1" EDID="D-20000119_002.e" ESNO="1" CDID="D-20000119_002.c"
CSNO="1" />
<SENT ORDER="2" EDID="D-20000119_002.e" ESNO="2" CDID="D-20000119_002.c"
CSNO="2" />
<SENT ORDER="3" EDID="D-20000119_002.e" ESNO="3" CDID="D-20000119_002.c"
CSNO="3" />
<SENT ORDER="4" EDID="D-20000119_002.e" ESNO="4" CDID="D-20000119_002.c"
CSNO="4" />
<SENT ORDER="5" EDID="D-20000119_002.e" ESNO="5" CDID="D-20000119_002.c"
CSNO="5" />
<SENT ORDER="6" EDID="D-20000119_002.e" ESNO="6" CDID="D-20000119_002.c"
CSNO="5" />
</SENTALIGN>
```

Figure A.7: Sentalign object

```
<?xml version='1.0'?>
<SENT-JUDGE QID="551">
<S DID="D-19980731_003.e" PAR="1" RSNT="1" SNO="1">
                <JUDGE N="smith" UTIL="10"/>
                <JUDGE N="huang" UTIL="10"/>
                <JUDGE N="moorthy" UTIL="6"/>
</S>
<S DID="D-19980731_003.e" PAR="2" RSNT="1" SNO="2">
                <JUDGE N="smith" UTIL="6"/>
                <JUDGE N="huang" UTIL="10"/>
                <JUDGE N="moorthy" UTIL="10"/>
</S>
<S DID="D-19980731_003.e" PAR="3" RSNT="1" SNO="3">
                <JUDGE N="smith" UTIL="6"/>
                <JUDGE N="huang" UTIL="9"/>
                <JUDGE N="moorthy" UTIL="10"/>
</S>

<S DID="D-19981105_011.e" PAR="5" RSNT="2" SNO="7">
                <JUDGE N="smith" UTIL="2"/>
                <JUDGE N="huang" UTIL="1"/>
                <JUDGE N="moorthy" UTIL="4"/>
</S>
</SENT-JUDGE>
```

Figure A.8: Sentjudge object

```
[1]The Disaster Relief Fund Advisory Committee has approved a
grant of $3 million to Hong Kong Red Cross for emergency relief
for flood victims in Jiangxi, Hunan and Hubei, the Mainland.
[2]Together with the earlier grant of $3 million to World Vision
Hong Kong, the Advisory Committee has so far approved $6 million from the
Disaster Relief Fund for relief projects to assist the victims
affected by the recent floods in the Mainland.
[3]The Disaster Relief Fund Advisory Committee has approved a
grant of $3 million to the Salvation Army for emergency relief
for flood victims in Hunan and Guangxi, the Mainland.
[4]The Disaster Relief Fund Advisory Committee has approved a
grant of $5.39 million to Medecins Sans Frontieres for emergency
relief for flood victims in Hunan, Sichuan and Yunnan, the Mainland.
[5]To ensure that the money will be used for the purpose
designated, the Government has required Medecins Sans Frontieres
to submit an evaluation report and audited accounts on the use of
the grant after the project has been completed.
[6]To ensure that the money will be used for the purpose
designated, the Government has required World Vision Hong Kong to
submit an evaluation report and audited accounts on the use of
the grant after the project has been completed.
[7]The Disaster Relief Fund Advisory Committee has approved a
grant of $3 million to the Hong Kong Committee for United Nations
Children's Fund (UNICEF) for emergency relief for flood victims
in Hubei, Hunan, Anhui, Heilongjiang, Jilin and Inner Mongolia
Autonomous Region, the Mainland.
[8]The Disaster Relief Fund Advisory Committee has approved a
grant of $1 million to Oxfam Hong Kong for relief for flood
victims in Shaanxi, Guangxi and Yunnan, the Mainland.
[9]Together with the earlier grants of $7 million to World Vision
Hong Kong, $3 million to Hong Kong Red Cross, $3 million to the
Salvation Army, $5.39 million to Medecins Sans Frontieres and $3
million to Hong Kong Committee for United Nations Children's
Fund, the Advisory Committee has now approved in
total $22.39 million from the Disaster Relief Fund for various
relief projects to assist the victims affected by the recent
floods in the Mainland.
[10]The Committee hopes that the grant can help to provide some
immediate relief to those affected.
[11]The Committee is concerned about the continuing hardship
brought about by floods and droughts in North Korea over the past
few years and hopes that the grant could help to provide some immediate
relief.
[12]Together with the earlier grants of $2.5 million to Medecins
Sans Frontieres Hong Kong in February 1998 and $1 million to
Cedar Fund Ltd. in April 1998, the Advisory Committee has
recently approved in total $5.5 million from the Disaster Relief
Fund for various relief programmes in North Korea.
[13]The Disaster Relief Fund Advisory Committee has approved a
grant of $1.5 million to World Vision Hong Kong for emergency
relief to flood victims in Bangladesh.
```

Figure A.9: Summary object

Config

The MEAD driver program requires a configuration file which specifies for it all of the programs and data it needs. See an example in Figure A.10.

```
<MEAD-CONFIG LANG="ENG" CLUSTER-PATH="/afs-clair/MEAD3/data/" \\
TARGET="Group_551 DATA-DIRECTORY=/afs-clair/MEAD3/data/">

<FEATURE-SET BASE-DIRECTORY="/afs-clair/MEAD3/data/feature/">
  <FEATURE NAME="Centroid"
SCRIPT="/afs-clair/MEAD3/programs/scripts/Centroid.pl HK-WORD-enidf ENG"/>
  <FEATURE NAME="Position"
SCRIPT="/afs-clair/MEAD3/programs/scripts/Position.pl"/>
   <FEATURE NAME="Length"
SCRIPT="/afs-clair/MEAD3/programs/scripts/Length.pl"/>
</FEATURE-SET>

<CLASSIFIER COMMAND-LINE="./hkmead.pl Centroid 1 Position 1 Length 9"
SYSTEM="MEADORIG" RUN="09/24"/>

<RERANKER COMMAND-LINE="./default-reranker.pl MEAD-cosine 0.7"/>

<COMPRESSION BASIS="sentences" PERCENT="20"/>

</THE-WORM-CONFIG>
```

Figure A.10: Mead Config object

<**MEAD-CONFIG**>
LANG: ENG or CHIN
CLUSTER-PATH: Path to the .cluster file you want to summarize.
DATA-DIRECTORY: Path where the source documents in docsent format are located.
TARGET: The name of the cluster file (without the .cluster)

<**FEATURE-SET**>
BASE-DIRECTORY: Path where MEAD will produce features.

<**FEATURE**> NAME: The name of the feature to use
SCRIPT: The full-path pointer to (including options) the script which will be used to generate this feature should it not exist in BASE-DIRECTORY above

<**CLASSIFIER**>
COMMAND-LINE: Should point to hkmead.pl (whereever it is).

<**RERANKER**>
COMMAND-LINE: Should point to default-reranker.pl (wherever it is).

<**COMPRESSION**>
BASIS: sentences or words
PERCENT: What percentage of the full document length should the summary document length be?

A.4 Installation

A.4.1 Downloading MEAD

Download MEAD from the workshop website (see above).

A.4.2 Installing MEAD

1. You need to have Perl installed. The English examples referred to in this documentation have been tested on Perl 5.6.0 on Solaris 5.7 and Linux (kernel 2.2 and 2.4). The Chinese examples have been tested on Linux (kernel 2.2) with perl 5.6.0.

2. Unpack MEAD 306.tar.gz. From now on the directory in which you have installed the MEAD files will be referred to as MEADBASE.

3. from MEADBASE, run "`perl Install.PL`". If there are installation-related problems in running MEAD later, refer to this script's output when emailing the MEAD team.

A.4.3 Running MEAD on the English Example

1. `cd MEADBASE/bin/program`

2. `cat mead.config | driver.pl \`

   `> English.extract`

   This config file is built for you automatically. See section A.3.2 for instructions on configuration files.

3. ```
   ../extensions/extract_to_summary.pl \
         ../../data/cluster/GA3 ../../data/ English.extract
   ```

A.4.4 Running MEAD on the Chinese Example

Preliminary Notes

We have provided routines for converting clusters of plain text Chinese documents into MEAD compatible data. The only stipulations we place on document formatting are as follows:

1. You should know the encoding of the documents you are working with. If you're not sure, a good rule of thumb is as follows:

   Simplified Chinese: GB2312

   Traditional Chinese: BIG5

2. All of the documents in the cluster should be encoded using the same standard (i.e. don't mix BIG5 and GB2312 documents).

3. The documents should be word-segmented. Note: We used the segmenter at http://www.mandarintools.com to segment the example. This segmenter is quite old, and we advise finding another one for best results.

List Format

If you wish MEAD to summarize your documents as a multidocument cluster, you should provide to us a file in the following format:

```
<pointer-to-file1>
<pointer-to-file2>
...
<pointer-to-filen>
```
```

GB18030 Compatability

As of the writing of this document, the glibc implementation of iconv, a library which converts among different encodings is NOT fully compatible with the latest encoding standard of the People's Republic of China (GB18030). This means that many documents (I find about 1/4 on xin hua wang) from up-to-date Chinese websites will crash the conversion routines.
NOTE: GB18030 is backwards compatible (all GB2312 encodings map to the same characters), so many documents that are actually encoded in GB18030 are labeled as GB2312 documents. If these documents contain a character which is undefined in GB2312, they will crash the conversion scripts.

System Compatibility

1. Linux: Fully compatible with GB2312. Compatible with SOME parts of GB18030.
2. Solaris 7 and below: I have NOT gotten these to work on GB2312. This will be addressed ASAP. To test your system try "`iconv -f gb2312 -t BIG5`".
3. Solaris 8 and above: I haven't had a chance to test these. Sun claims that Solaris 8 (02/02 patch) and above are fully compatible with GB18030.

Running The Example

This example is a two-article cluster from xin hua wang (The website of China's largest news agency). It discusses Taiwan's decision to use "Common pinyin". It is encoded in GB2312.

1. Run the conversion script.

```
<LINUX USERS>
$cd <MEAD_BASE>/bin/extensions
$make-CHIN-docsent chin-example/commonpy.list GB2312
$cd ../program

<SOLARIS USERS>
$cd <MEAD_BASE>/bin/extensions
$make-CHIN-docsent chin-example/commonpy.list gb2312
$cd ../program
```

2. Edit the mead config file

```
<Change the cluster>
Replace target="GA3" with target="commonpy.list"

<Change the Language>
Replace Lang="ENG" with Lang="CHIN"
Replace "ENG" with CHIN" in the Centroid feature script

<Change the IDF database>
Replace "enidf" with "cnidf" in the "Centroid" feature script.
Replace "enidf" with "cnidf" in the "Reranker" command line.
```

3. Run

```
$cat mead.config | driver.pl > commonpy.extract
$../extensions/extract_to_summary.pl \
../../data/cluster/commonpy.list.cluster ../../data/docsent/ \
commonpy.extract >! commonpy.summary
```

A.5 Creating new Feature Scripts

A.5.1 Introduction to MEAD Features

MEAD extractive summaries score sentences according to certain features these sentences have. The default classifier (HKMEAD) uses Position, Centroid, and Length, but MEAD features can potentially refer to any feature that a sentence has (how many named entities or anaphora it contains, for instance). The only stipulation that MEAD places on its features is that they be real-valued. The MEAD distribution contains also the FirstSim feature which computes the cosine similarity between a sentence and the first sentence in the document.

In order to facilitate the easy creation and integration of new features, MEAD provides an interface to the features using the interface library Feature_Extractor.pm. This section describes the use of this library.

A.5.2 The Feature Extractor Interface

Sent-Feature Files

Sent-Feature files contain the values of features for each sentence. These are the output of all Feature Calculation scripts. Below is a DTD for sent-feature files:

```
<!ELEMENT SENT-FEATURE (S)*>

<!ELEMENT S (FEATURE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT FEATURE EMPTY>
<!ATTLIST FEATURE
    N CDATA #REQUIRED
    UTIL CDATA #REQUIRED>
```

Figure A.11: Sentfeature.dtd

A feature script that uses Feature_Extractor does not need to explicitly write Sent-Feature files, however. The library will do this for you.

Three-Pass Feature Calculation

Feature Calculation is done in three stages: Cluster, Document, and Sentence. A Feature Script **must** use the **Sentence** stage. The other two stages are optional. In order to implement the processing necessary for a stage, write a subroutine *sub-x* which corresponds to the stage and pass a hash with a key of ”stage” and a value of a reference to *sub-x* into the Feature_Extractor library function *Do*. A simple example follows:
Note that the ’Sentence’ string must appear verbatim (case-sensitive) as the key of the hash entry whose value is the reference to the ’Sentence’ subroutine. A call which specifies all stages follows (again the string keys must match exactly and are case-sensitive):

```
Do(\{'Cluster'=>\&cluster, 'Document'=>\&document, 'Sentence' =>\&sentence}\);
```

The $datadir variable points to the directory containing the docsent files whose sentences you want to calculate the features for.

The Cluster Stage

Cluster routines are passed a cluster and can do what processing they need to with that cluster. They are called once per cluster.

```
use strict;

use Feature_Extractor;

my $datadir = shift;

Do($datadir, {'Sentence' =>\&sentence});

sub sentence {}
```

Figure A.12: Sample use of a feature during the Sentence Stage

- Clusters are references to hashes whose keys are DIDs (strings) and whose values are references to Documents
- Documents are arrays of Sentences
- Sentences are hashes whose keys are features (strings) and whose values are the values of those features. The important features follow:

  "TEXT" (string) The text of the sentence

  "DID" (string) The DID of the document to which the sentence belongs

  "SNO" (string) The number of the sentence in its document

The Document Stage

Document routines are passed a Document and can do what processing they need to with that Document. They are called once for each document in the cluster.

- Documents are arrays of Sentences (See above for a description of a Sentence)

The Sentence Stage

Sentence routines are passed two variables: A Sentence and a reference to a feature_vector. Sentences are described in the "Cluster Stage" section above. Feature_vectors are hashes whose keys are the names of features (strings) and whose values are the real values of the features named by those strings. For example:

```
{'Centroid'=>0.2, 'Position' =>1}
```

After the Sentence routine has been called for every sentence in every document in the cluster, the Feature_extractor library writes to standard out a Sent-Feature file containing the values for the features specified in the feature_vector for each sentence.

A Skeleton Feature Extraction Routine

Included with the MEAD distribution is **MEAD-BASE/programs/scripts/Skeleton.pl**, which is a routine that provides a minimal feature calculation and can be used as a jumping-off point to write your own features. If Feature_Extractor is installed correctly, then the following command should produce a sent-feature file identical to **GA3.skeleton.sentfeature** (also in **MEAD-BASE/programs/scripts/**):

```
echo 'MEAD_BASE/data/GA3.cluster' | Skeleton.pl
```

A.6 Adding new features to the classifier

The ranker computes scores for each sentence.

- Input: This is a feature file (usually the output of fcombine.pl). Every feature file specifies for each sentence in a cluster, a set of features and a value for each feature (for that sentence).
- Output: This is a sentjudge file. It indicates a real number for each sentence in a cluster. In the case of a classifier, this real number indicates a score for each sentence.

A.6.1 command line arguments

The COMMAND-LINE attribute of CLASSIFIER should read ”hkmead.pl <Feature1> val1 ... <Featuren> valn”. Each sentence receives a score that is a linear combination of the features listed (provided they are in the input feature file) EXCEPT for the ”Length” feature. Thus each Feature should be given with the coeficcient of that feature’s ”dimension” in the linear combination.

”Length”, if it is given, is a cutoff feature. Any sentence with a length shorter than ”Length” is automatically given a score of 0, regardless of its other features. ”Length” is the only feature that has these semantics.

Thus ./hkmead.pl Centroid 2 Position 0.5 Length 12 has the following interpretation:

$$score(sentence) = \left\{ \begin{array}{rcl} 2 \cdot (Centroid) + 0.5 \cdot (Position) & : & Length(s) > 12 \\ 0 & : & Length(s) < 12 \end{array} \right\}$$

A.7 Adding new relations (sentence reranker)

The reranker is used to reassign scores to sentences based on relationships between pairs of sentences. For example, it can be used to give lower scores to repeated sentences or higher scores to sentences that are in an anaphoric relationship with another sentence.

- Input: This is a reranker-info file (the dtd is in the ”dtd/” directory). A reranker-info file has three components:
  1. Compression information –> PERCENT specifies the percentage compression (ie 20=20% compression), BASIS specifies the % granularity at which to measure compression. It should be either ”words” or ”sentences”
  2. Cluster information –> This is a ”.cluster” file
  3. Sentjudge information –> This is a ”.sentjudge” file
- Output: The reranker, like the classifier, outputs a sentjudge file. This way you can (if you want) have no reranker at all.

A.7.1 command line arguments

The COMMAND-LINE attribute of RERANKER should read ”default-reranker.pl <Similarity-Function> <Threshold-Value>”.

<Similarity Function>: The SimRoutines library specifies a has with strings as keys and references to functions as values. This argument is such a string (key in this hash). The reranker will use it to calculate a similarity value for two sentences.

<Threshold Value>: If sentences are ordered 1, 2, ..., n by score, for a sentence s_j, if the similarity value is less than this threshold for all $s_i, 0 < i < j$, the reranker will add 1000 to the score of j. Otherwise, it will do nothing (effectively ranking the sentence last). Thus the command line ”default-reranker.pl MEAD-cosine 0.7” says:

When comparing sentences in the above fashion, use the MEAD-cosine similarity routine. If this routine returns a value greater than 0.7 for a pair of sentences, do not add 1000 to the lower-scoring sentence.

A.8 SVM Documentation

This section describes the data format and instructions for training and evaluation for sentence extraction in MEAD using Support Vector Machines(SVM).

A.8.1 Data Format

The format of training, tuning (development), and testing data are similar. The format is also similar to the data format expected by the SVM package. Each data file contains cases or samples. Each sample corresponds to a sentence and its feature values. Each sample is described by one line of record with syntax as follows:

<class> <feature-id1>:<feature-value1> <feature-id2>:<feature-value2> ...

<class> can be 1 or -1 representing the corresponding sentence is included or not included in the sample summary.

<feature-idx> is an integer representing a feature id.

<feature-valuex> is a real number representing a feature value.

Therefore, each record contains those features and their corresponding values for a particular sentence. It also contains whether or not the sentence is included in the sample summary.

Note that the feature values should be normalized so that the values fall between 0 and 1.

A.8.2 Instructions for Porting, Training and Evaluation

Porting

- Make a directory, e.g. trainable_mead which will contain all the data files and SVM package.
- Download SVM package
- Copy svm_classify.c to replace the original svm_classify.c (save a backup of the original svm_classify.c as advised)
- Compile the SVM package
- Prepare the training, tuning(development), and testing data. Follow the data format described above. (Note that the feature values should be normalized.)

Training

%**SVM/svm_learn -j <cost-parameter> <training.data> <learned-model>**
where:
<cost-parameter>is a parameter by which training errors on positive examples outweight errors on negative examples (default 1)
<training.data> is the training data set
<learned-model> is the output learned model
e.g.
%**SVM/svm_learn -j 5 training.data learned-model-j5**

The above command invoke the training process using the training data (training.data)with cost parameter 5.

The output of the learned model is stored in the file learned-model-j5. This learned-model will be used in the tuning and evaluation stages

Tuning (Development)

% **SVM/svm_classify <train.dev.data> <learned-model>**
where:
<train.dev.data> is the tuning (development) data set
<learned-model> is the learned model obtained from the training stage
e.g.

% **SVM/svm_classify train.dev.data learned-model-j5**

This command invokes the classification process on the tuning data - train.dev.data using the learned model - learned-model-j5. The linear weights of each feature are displayed. The accuracy, precision, and recall metrics are also shown.

Typically, one will conduct training using different parameters such as different cost factors. Then invoke the classification process for each learned model. One can choose the desired model based on a particular metric such as recall.

Testing

% **SVM/svm_classify <testing.data> <learned-model>**
where:
<testing.data> is the testing data <learned-model> is the selected learned model after tuning
e.g.
% **SVM/svm_classify testing.data learned-model-j5**
This command invokes the classification process on the testing data - testing.data

A.9 Miscellaneous tools

A number of tools comes with the MEAD distribution. They can be found it the ”extensions” directory.

A.9.1 mkconfig

extensions/mkconfig.pl

A.9.2 Random and Lead-based single-document summarizers

extensions/lead-based.pl extensions/random-based.pl

A.9.3 Random and Lead-based multi-document summarizers

extensions/cluster-random-extractor.pl

A.10 Evaluation

The MEAD evaluation toolkit (which implements precision, recall, kappa, cosine, unigram and bigram overlap, and relative utility) is available at http://perun.si.umich.edu/clair/meadeval

A.11 Project Web site

The MEAD project has a Web page at Johns Hopkins University.

Figure A.13: Web site for the MEAD projects

A.12 Frequently Asked Questions

A.12.1 Does MEAD only work on the HK News Corpus?

No. The example above in Section Installation shows how to use a different corpus.

A.12.2 Can I contribute to MEAD?

Sure. Please send mail to the MEAD mailing list: mead@majordomo.si.umich.edu

A.12.3 How can I get help?

Please refer to the MEAD homepage for help.

http://www.clsp.jhu.edu/ws2001/groups/asmd

A.12.4 Do I need a license to use MEAD

Not for the moment. Once we are beyond the beta stage, we will look into this issue.

A.13 Demos

- www.newsinessence.com
- perun.si.umich.edu/clair/meaddemo

A.14 Credits for MEAD

- Dragomir Radev - MEAD 1.0 (2000),
- Sasha Blair-Goldensohn - MEAD 2.0 (Spring 2001),
- John Blitzer, Elliott Drabek, Arda Çelebi, Hong Qi, Dragomir Radev, Simone Teufel, Horacio Saggion, Wai Lam, Danyu Liu, Sanjeev Khudanpur - MEAD 3.0 (the current version, Summer and Fall 2001),
- Inderjeet Mani, Chin-Yew Lin - project affiliates,
- Michael Topper - documentation, demos, and porting,
- Adam Winkel - demos,
- Arda Çelebi - Web site and distribution,
- Fred Jelinek, Bill Byrne, Sanjeev Khudanpur, Laura Graham, Jacob Laderman - hosts of the summer workshop at Johns Hopkins where MEAD 3.0 was developed,
- Stephanie Strassel, Chris Cieri, David Graff (all from LDC) - corpus creation and annotation,
- Ralph Weischedel, Regina Barzilay, David Day, Greg Silber, Dan Melamed, Sean Boisen - miscellaneous advice and resources, and finally,
- The MEAD beta testers, especially John Murdie

A.15 XML DTDs

A.15.1 cluster.dtd

```
<!ELEMENT CLUSTER (D)*>
<!ATTLIST CLUSTER
   LANG (CHIN|ENG) "ENG">

<!ELEMENT D EMPTY>
<!ATTLIST D
   DID   ID    #REQUIRED
   ORDER CDATA #IMPLIED>
```

A.15.2 docjudge.dtd

```
<!ELEMENT DOC-JUDGE (D)*>
<!ATTLIST DOC-JUDGE
     QID    CDATA #REQUIRED
     SYSTEM CDATA #REQUIRED
     LANG   (CHIN|ENG) "ENG">

<!-- LANG refers to the language of the retrieval process.
     Thus, it is the language of the documents.
     However, the original language of the query might be
     different.
     Look this up in QID. -->

<!ELEMENT D EMPTY>
<!ATTLIST D
     DID   ID     #REQUIRED
     RANK  CDATA  #IMPLIED
     CORR-DOC CDATA #IMPLIED
     SCORE CDATA  #REQUIRED>
```

A.15.3 docpos.dtd

```
<!--  DTD for POS tagged text  -->

<!ELEMENT DOCPOS (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCPOS
   DID   CDATA      #REQUIRED
   DOCNO CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (S)*>
<!ELEMENT TEXT (S)*>

<!ELEMENT S (W)*>
<!ATTLIST S
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>
    <!-- PAR: paragraph no
         RSNT: relative sentence no (within paragraph)
         SNO: absolute sentence no -->

<!ELEMENT W (#PCDATA)>
<!ATTLIST W
    C   CDATA #REQUIRED
    L   CDATA #IMPLIED>

<!-- C is the POS category. L is the lemma -->
```

A.15.4 docsent.dtd

```
<!--  DTD for sentence-segmented text  -->

<!ELEMENT DOCSENT (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCSENT
   DID   CDATA      #REQUIRED
   DOCNO CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (S)*>
<!ELEMENT TEXT (S)*>

<!ELEMENT S (#PCDATA)>
<!ATTLIST S
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>
    <!-- PAR: paragraph no
         RSNT: relative sentence no (within paragraph)
         SNO: absolute sentence no -->
```

A.15.5 document.dtd

```
<!--  DTD for original, non-segmented text  -->

<!ELEMENT DOCUMENT (EXTRACTION-INFO?, BODY)>
<!ATTLIST DOCUMENT
   DID   CDATA      #REQUIRED
   DOCNO   CDATA      #IMPLIED
   LANG  (CHIN|ENG) "ENG"
   CORR-DOC CDATA   #IMPLIED>
   <!-- DID : documentid
       LANG: language    -->

<!ELEMENT EXTRACTION-INFO EMPTY>
<!ATTLIST EXTRACTION-INFO
    SYSTEM  CDATA  #REQUIRED
    RUN     CDATA  #IMPLIED
    COMPRESSION CDATA #REQUIRED
    QID     CDATA  #REQUIRED>

<!ELEMENT BODY (HEADLINE?,TEXT)>

<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
```

A.15.6 extract.dtd

```
<!ELEMENT EXTRACT (S)*>
<!ATTLIST EXTRACT
   QID         CDATA #REQUIRED
   COMPRESSION CDATA #REQUIRED
   SYSTEM      CDATA #REQUIRED
   JUDGE       CDATA #IMPLIED
   JUDGENO     CDATA #IMPLIED
   RUN         CDATA #IMPLIED
   SENTS_TOTAL CDATA #IMPLIED
   WORDS_TOTAL CDATA #IMPLIED
   LANG        CDATA #REQUIRED>

<!ELEMENT S EMPTY>
<!ATTLIST S
   ORDER       CDATA #REQUIRED
   DID         CDATA #REQUIRED
   SNO         CDATA #IMPLIED
   PAR         CDATA #IMPLIED
   RSNT        CDATA #IMPLIED
   UTIL        CDATA #IMPLIED>
```

A.15.7 query.dtd

```
<!ELEMENT QUERY (TITLE,DESCRIPTION?,NARRATIVE?)>
<!ATTLIST QUERY
   QID  CDATA #REQUIRED
   QNO  CDATA #REQUIRED
   LANG (CHIN|ENG) "ENG"
   TRANSLATED (YES|NO) "NO"
   ORIGLANG (CHIN|ENG) "CHIN"
   TRANS-METHOD (AUTO|MAN) "AUTO">

<!-- QID: unique query no, eg. 125-CA or 125-E
     QNO: LDC query no for content, eg. 125
     LANG: of query
     TRANSLATED: is it an original query or not?
     ORIGLANG: If translated, from which language (from the other
     one, of course
!)
     TRANS-METHOD: Automatically translated or manually? -->

<!ELEMENT TITLE       (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT NARRATIVE   (#PCDATA)>
```

A.15.8 reranker-info.dtd

```
<!-- DTD for input to rerankers -->

<!ELEMENT RERANKER-INFO (COMPRESSION, CLUSTER, SENT-JUDGE)

<!ELEMENT COMPRESSION EMPTY>
<!ATTLIST COMPRESSION
    PERCENT CDATA #REQUIRED
    BASIS   CDATA #REQUIRED
>

<!ELEMENT CLUSTER (D)*>
<!ATTLIST CLUSTER
   LANG (CHIN|ENG) "ENG">

<!ELEMENT D EMPTY>
<!ATTLIST D
   DID   ID    #REQUIRED
   ORDER CDATA #IMPLIED>

<!ELEMENT SENT-JUDGE (S)*>
<!ATTLIST SENT-JUDGE
    QID  CDATA  #REQUIRED>

<!ELEMENT S (JUDGE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT JUDGE EMPTY>
<!ATTLIST JUDGE
    N    CDATA #REQUIRED
    UTIL CDATA #REQUIRED>
```

A.15.9 sentalign.dtd

```
<!ELEMENT SENTALIGN (SENT+)>
<!ATTLIST SENTALIGN
   ENG         CDATA #REQUIRED
   CHI         CDATA #REQUIRED
   LANG        CDATA #REQUIRED>

<!ELEMENT SENT EMPTY>
<!ATTLIST SENT
   ORDER       CDATA #REQUIRED
   EDID        CDATA #REQUIRED
   ESNO        CDATA #REQUIRED
   CDID        CDATA #REQUIRED
   CSNO        CDATA #REQUIRED>

<!-- ORDER: the pairwise number
     EDID:  english document name
     ESNO:  english sentence number
     CDID:  chinese document name
     CSNO:  chinese sentence number -->
```

A.15.10 sentjudge.dtd

```
<!ELEMENT SENT-JUDGE (S)*>
<!ATTLIST SENT-JUDGE
    QID  CDATA  #REQUIRED>

<!ELEMENT S (JUDGE)*>
<!ATTLIST S
    DID  CDATA #REQUIRED
    PAR  CDATA #REQUIRED
    RSNT CDATA #REQUIRED
    SNO  CDATA #REQUIRED>

<!ELEMENT JUDGE EMPTY>
<!ATTLIST JUDGE
    N    CDATA #REQUIRED
    UTIL CDATA #REQUIRED>
```

A.15.11 the-worm-config.dtd

```
<!ELEMENT THE-WORM-CONFIG (FEATURE-SET, CLASSIFIER, RERANKER,
COMPRESSION) >
<!ATTLIST THE-WORM-CONFIG
   LANG CDATA #REQUIRED
   CLUSTER-PATH CDATA #IMPLIED
   DATA-DIRECTORY CDATA #IMPLIED
   TARGET CDATA #IMPLIED >

<!ELEMENT FEATURE-SET (FEATURE*) >
   BASE-PATH CDATA #IMPLIED >

<!ELEMENT FEATURE EMPTY >
<!ATTLIST FEATURE
   FEATURE CDATA #REQUIRED >

<!ELEMENT CLASSIFIER EMPTY >
<!ATTLIST CLASSIFIER
   COMMAND-LINE CDATA #REQUIRED
   SYSTEM CDATA #IMPLIED
   RUN CDATA #IMPLIED >

<!ELEMENT RERANKER EMPTY >
<!ATTLIST RERANKER
   COMMAND CDATA #REQUIRED
   BASIS (sentences|words) #REQUIRED
   PERCENT CDATA #REQUIRED >
```

Bibliography

Aho, Alfred, Shih-Fu Chang, Kathleen McKeown, Dragomir Radev, John Smith, and Kazi Zaman. 1997. Columbia Digital News System : An Environment for Briefing and Search over Multimedia Information. In *Proceedings of IEEE International Conference on the Advances in Digital Libraries*. Washington, DC.

Allan, James, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal Summaries of News Topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*.

Alterman, R. 1992. Text Summarization. In S.C. Shapiro, ed., *Encyclopedia of Artificial Intelligence*, vol. 2, 1579–1587. Jonh Wiley & Sons, Inc.

Alterman, R., and L.A. Bookman. 1990. Some Computational Experiments in Summarization. *Discourse Processes* 13: 143–174.

Ando, Rie, Branimir Boguraev, Roy Byrd, and Mary Neff. 2000. Multi-document summarization by visualizing topical content. In *Proceedings of the Workshop on Automatic Summarization, ANLP-NAACL2000*. Seattle, WA.

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

Aone, Chinatsu, Mary Ellen Okurowski, James Gorlinsky, and Bjornar Larsen. 1997. A scalable summarization system using robust NLP. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 66–73.

Aone, Chinatsu, Mary Ellen Okurowski, James Gorlinsky, and Bjornarn Larsen. 1999. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In I. Mani and M.T. Maybury, eds., *Advances in Automatic Text Summarization*, 71–80. The MIT Press.

Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison Wesley.

Baldwin, Breck, and Thomas S. Morton. 1998. Dynamic coreferencebased summarization. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*.

Ballesteros, Lisa, and W. Bruce Croft. 1998. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, 64–71.

Barzilay, Regina, and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 10–17. Madrid, Spain.

Barzilay, Regina, Kathleen McKeown, and Michael Elhadad. 1999. Information Fusion in the Context of Multi-Document Summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 550–557. Maryland, USA.

Baxendale, P.B. 1958. Man-made Index for Technical Litterature - an experiment. *IBM J. Res. Dev.* 2(4): 354–361.

BBN. 2000. *IdentiFinder: User Manual*. GTE. BBN Technologies. Version 5.0.

Benbrahim, M., and K. Ahmad. 1995. Text Summarisation: the Role of Lexical Cohesion Analysis. *The New Review of Document & Text Management* 321–335.

Berger, Adam L., and Vibhu O. Mittal. 2000a. OCELOT: a system for summarizing Web pages. In *Research and Development in Information Retrieval*, 144–151.

Berger, Adam L., and Vibhu O. Mittal. 2000b. Query-relevant Summarization using FAQs. In *Proceedings of the 38th Annual Meeting of the Association for Computation Linguistics*, 294–301. Association for Computational Linguistics (ACL).

Boguraev, Branimir, and Chris Kennedy. 1997. Salience based content characterization of text documents. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 2–9.

Boguraev, Branimir, Chris Kennedy, Rachel Bellamy, Sascha Brawer, Y. Wong, and J. Swartz. 1998. Dynamic presentation of document content for rapid on-line skimming. In *AAAI Spring 1998 Symposium on Intelligent Text Summarization*.

Borko, H., and C. Bernier. 1975. *Abstracting Concepts and Methods*. Academic Press.

Borko, Harold, and Seymour Chatman. 1963. Criteria for acceptable abstracts: A survey of abstractors' instructions. *American Documentation* 14(2): 149–160.

Brandow, Ron, Karl Mitze, and Lisa F. Rau. 1995. Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing and Management* 31(5): 675–685.

Brown, Ann L., and Jeanne D. Day. 1983. Macrorules for summarizing text: The developments of expertise 22: 1–14.

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning Sentences in Parallel Corpora. In *Meeting of the Association for Computational Linguistics*, 169–176.

Buckley, Chris. 1985. Implementation of the SMART Information Retrieval System. Technical Report TR85-686.

Carbonell, Jaime G., and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Research and Development in Information Retrieval*, 335–336.

Carbonell, Jaime G., Yiming Yang, Robert E. Frederking, Ralf D. Brown, Yibing Geng, and Danny Lee. 1997. Translingual Information Retrieval: A Comparative Evaluation. In *IJCAI (1)*, 708–715.

Carletta, Jean. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics* 22(2): 249–254.

Chen, H.H., G.W. Bian, and W.C. Lin. 1999. Resolving translation ambiguity and target polysemy in cross-language information retrieval. In *ACL 99*, 215–222.

Cole, R.E., ed. 1995. *Survey of the State of the Art in Human Language Technology*, chapter 13, 475–518. Cambridge University Press.

Cremmins, Edward. 1993. Valuable and Meaningful Text Summarization in Thoughts, Words, and Deeds. In Brigitte Endres-Niggemeyer, Jerry Hobbs, and Karen Sparck Jones, eds., *Summarising Text for Intelligent Communication*. Dagstuhl, Germany.

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

Crochemore, Maxime, and Wojciech Rytter. 1994. *Text algorithms*. Oxford University Press.

Cuts, Short. 1994. Science and technology section. *The Economist* 17: 85–86.

Deerwester, Scott C., 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.

DeJong, Gerald Francis. 1979. Skimming Stories in Real Time: an Experiment in Integrated Understanding. Technical Report 158, New Haven, CT.

DeJong, Gerald Francis. 1982. An Overview of the FRUMP System. In W.G. Lehnert and M.H. Ringle, eds., *Strategies for Natural Language Processing*, 149–176. Lawrence Erlbaum Associates, Publishers.

Donaway, Robert L., Kevin W. Drummey, and L.A. Mather. 2000. A Comparison of Rankings Produced by Summarization Evaluation Measures. In *Proceedings of the Workshop on Automatic Summarization, ANLP-NAACL2000*, 69–78. Association for Computational Linguistics.

DUC2000. 2000. *Proceedings of the Workshop on Text Summarization*.

Edmundson, H.P. 1964. Problems in automatic extracting. *Communications of the Association for Computing Machinery* 7: 259–263.

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

Endres-Niggemeyer, Brigitte. 1993. A Naturalistic Models of Abstracting. In *Preprints of Summarizing Text for Intelligent Communication. Dagstuhl Seminar Report 79*, 21–25. Schloss Dagstuhl, Germany.

Endres-Niggemeyer, Brigitte, Jerry Hobbs, and Karen Sparck Jones. 1993. Summarizing text for Intelligent Communication. Technical report, Dagstuhl. In Dagstuhl Seminar Report, IBFI GmbH, Schloss Dagstuhl, Wadern, Germany.

Endres-Niggemeyer, Brigitte, E. Maier, and A. 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.

Fellbaum, Christiane, ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Firmin, Thérèse, and Michael J. Chrzanowski. 1999. An Evaluation of Automatic Text Summarization Systems. In I. Mani and M.T. Maybury, eds., *Advances in Automatic Text Summarization*, 325–336.

Fukumoto, F., Y. Suzuki, and J. Fukumoto. 1997. An automatic extraction of key paragraphs based on context dependency. In *Proceedings of the 5th International on Applied Natural Language Processing, ANLP1997*. Washington.

Fum, D., G. Guida, and C. Tasso. 1985. Evaluating importance: A step towards text summarization. In *Proceedings of IJCAI*, 840–844.

Gaizauskas, R., T. Wakao, K Humphreys, H. Cunningham, and Y. Wilks. 1995. Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, California.

Gaizauskas, Robert, Kevin Humphreys, Saliha Azzam, and Yorick Wilks. 1997. Concepticons vs. Lexicons: An Architecture for Multilingual Information Extraction. In M.T. Pazienza, ed., *Information Extraction.A multidisciplinary Approach to an Emerging Information Technology. Lectures Notes in Artificial Intelligence*, vol. 1299, 28–43. Springer.

Gale, William A., and Kenneth W. Church. 1993. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics* 19(1): 75–102.

Garner, Ruth. 1982. Efficient text summarization: costs and benefits. *Journal of Education Research* 75: 275–279.

Gladwin, Philip, Stephen Pulman, and Karen Sparck Jones. 1991. Shallow processing and automatic summarizing: A first study. Technical Report Technical Report No. 223, University of Cambridge Computer Laboratory.

Goldstein, Jade, Mark Kantrowitz, Vibhu O. Mittal, and Jaime G. Carbonell. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Research and Development in Information Retrieval*, 121–128. Berkeley, California.

Goldstein, Jade, Vibhu O. Mittal, Jaime G. Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL workshop on Automatic Summmarization*. Seattle, WA.

Gong, Yihong, 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*.

Grefenstette, Gregory. 1998a. *Cross-Language Information Retrieval*. U.S.A.: Kluwer Academic Publishers.

Grefenstette, Gregory. 1998b. *Cross-Language Information Retrieval*, chapter The Problem of Cross-Language Information Retrieval, 1–9. Kluwer Academic Publishers.

Grefenstette, Gregory. 1998c. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Intelligent Text Summarization Symposium*, 111–117. Standford (CA), USA.

Grishman, Ralph, and Beth Sundheim. 1996. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics*. Copenhagen.

Grover, Claire, Andrei Mikheev, and Colin Matheson. 1999. LT TTT Version 1.0: Text Tokenisation Software. Technical report, Human Communication Research Centre, University of Edinburgh. `http://www.ltg.ed.ac.uk/software/ttt/index.html`.

Hahn, Udo. 1990. Topic Parsing: Accounting for Text Macro Structures in Full-Text Analysis. *Information Processing & Management* 26(1): 135–170.

Hallyday, M.A.K. and Hasan, Ruqaia. 1996. *Cohesion in English*. London: Longmans.

Hand, Thérèse F. 1997. A proposal for task-based evaluation of text summarization systems. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, 31–38. Madrid, Spain.

Hovy, Eduard, and Chin-Yew Lin. 1999. Automated Text Summarization in SUMMARIST. In I. Mani and M.T. Maybury, eds., *Advances in Automatic Text Summarization*, 81–94. The MIT Press.

Humphreys, Kevin, Robert Gaizauskas, and Hamish Cunningham. 2000. LaSIE Technical Specifications. Technical report, Department of Computer Science. University of Sheffield.

Hutchins, John. 1987. Summarization: Some Problems and Methods. In K.P. Jones, ed., *Meaning: The Frontier of Informatics*, vol. 9, 151–173. Aslib.

Hwang, Dosam, and Makoto Nagao. 1994. Aligning of Japanese and Korean texts by analogy. *94-NL-99* 94(9): 87–94.

ISO. 1976. Documentation—Abstracts for Publication and Documentation. ISO 214-1976. Technical report, International Organisation for Standardisation.

Jacobs, Paul S., and Lisa F. Rau. 1990. SCISOR: Extracting Information from On-line News. *Communications of the ACM* 33(11): 88–97.

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

Jing, Hongyan, and Kathleen McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In M. Hearst, Gey. F., and R. Tong, eds., *Proceedings of SIGIR'99. 22nd International Conference on Research and Development in Information Retrieval*, 129–136. University of California, Beekely.

Jing, Hongyan, and Kathleen McKeown. 2000. Cut and Paste Based Text Summarization. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 178–185. Seattle, Washington, USA, April 29 - May 4.

Jing, Hongyan, Kathleen McKeown, Regina Barzilay, and Michael Elhadad. 1998. Summarization Evaluation Methods: Experiments and Analysis. In *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06*, 60–68. Standford (CA), USA: The AAAI Press.

Johnson, F.C., Chris D. Paice, W.J. Black, and A.P. Neal. 1993. The application of linguistic processing to automatic abstract generation. *Journal of Document and Text Management* 1(3): 215–239.

Jones, P.A., and C.D. Paice. 1992. A 'select and generate' approach to to automatic abstracting. In A.M. McEnry and C.D. Paice, eds., *Proceedings of the 14th British Computer Society Information Retrieval Colloquium*, 151–154. Springer Verlag.

Kay, Martin, and Martin Roscheisen. 1993. Text-translation alignment. *Computational Linguistics* 19(1): 121–142.

Kintsch, W., and Teun A. van Dijk. 1975. Comment on se rappelle et on résume des histoires. *Langages* 40: 98–116.

Kintsch, Walter, and Teun A. van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review* 85(5): 363–394.

Knight, Kevin, and Daniel Marcu. 2000. Statistics-Based Summarization — Step One: Sentence Compression. In *Proceeding of The 17th National Conference of the American Association for Artificial Intelligence (AAAI-2000)*, 703–710.

Krippendorff, Klaus. 1980. *Content Analysis: An Introduction to its Methodology*. Beverly Hills, CA: Sage Publications.

Kupiec, Julian, Jan O. Pedersen, and Francine Chen. 1995. A Trainable Document Summarizer. In *Research and Development in Information Retrieval*, 68–73.

Kwok, K.L., N. Grunfeld, N. Dinstl, and M. Chan. 2000. TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS. In *The Ninth Text REtrieval Conference TREC 9*.

Ladas, Horald. 1997. Summarising research: A case study. Review of al Issue on Empirical Studies in Discourse Interpretation and Generation.

Lam, Wai, and Chao Y. Ho. 1998. Using a generalized instance set for automatic text categorization. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, eds., *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, 81–89. Melbourne, AU: ACM Press, New York, US.

Lam, Wai, C.K. Keung, and C.X. Ling. 2001a. *Instance Selection and Construction for Data Mining, Eds. Liu, H. and Motoda, H.*, chapter Learning Via Prototype Generation and Filtering. Kluwer Academic Publishers.

Lam, Wai, and K.Y. Lai. 2001. A Meta-Learning Approach for Text Categorization. In *ACM SIGIR*. To appear.

Lam, Wai, and J. Mostafa. 2001. Modeling User Interest Shift Using A Bayesian Approach. *Journal of American Society of Information Science and Technology, JASIST* 52(5): 416–429.

Lam, Wai, M. Ruiz, and P. Srinivasan. 1999. Automatic Text Categorization and Its Application to Text Retrieval. *IEEE Transactions on Knowledge and Data Engineering* 11: 865–879.

Lam, Wai, K.F. Wong, and C.Y. Wong. 2001b. Chinese Document Indexing Based on a New Partitioned Signature File: Model and Evaluation. *Journal of American Society of Information Science and Technology, JASIST* 52(7): 584–591.

Lam-Adesina, Adenike, and Gareth Jones. 2001. Applying Summarization Techniques for Term Selection in Relevance Feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*.

Lancaster, Frederick Wilfrid. 1998. *Indexing and Abstracting in Theory and Practice*. London, UK: Library Association.

Landis, J.R., and G.G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33: 159–174.

Langlais, Philippe. 1997. A System to Align Complex Bilingual Corpora. In *TMH-QPSR 4/1997*. KTH, Stockholm, Sweden.

Lehmam, Aberrafih. 1995. Le résumé des textes techniques et scientifiques, aspects linguistiques et computationnels. Ph.D. thesis, Universite de Nancy 2.

Lehnert, Wendy. 1981. Plot Units and Narrative Summarization. *Cognitive Science* 5(4): 293–331.

Lin, Chin-Yew. 1999. Training a Selection Function for Extraction. In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM)*, 55–62. ACM, Kansas City.

Lin, Chin-Yew, and Eduard Hovy. 1997. Identifying Topics by Position. In *Fifth Conference on Applied Natural Language Processing*, 283–290. Association for Computational Linguistics.

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

Maizell, R.E., J.F. Smith, and T.E.R. Singer. 1971. *Abstracting Scientific and Technical Literature*. Wiley-Interscience, A Division of John Wiley & Son, Inv.

Mani, Inderjeet. 2001. *Automatic Summarization*. Amsterdam/Philadephia: John Benjamins Publishing Company.

Mani, Inderjeet, and Eric Bloedorn. 1997a. Multi-document Summarization by Graph Search and Matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 622–628. American Association for Artificial Intelligence, Providence, Rhode Island.

Mani, Inderjeet, and Eric Bloedorn. 1997b. Summarizing similarities and differences among related documents. In *Proceedings of the 5th RIAO Computer Assisted Information Searching on Internet*, 373–387. Montreal, Canada.

Mani, Inderjeet, and Eric Bloedorn. 1999. Summarizing similarities and differences among related documents. *Information Retrieval* 1(1): 35–67.

Mani, Inderjeet, and Eric Bloedorn. 2000. Summarizing Similarities and Differences Among Related Documents. *Information Retrieval* 1(1).

Mani, Inderjeet, Kristian Concepcion, and Linda van Guilder. 2000. Using summarization for automatic briefing generation. In *Proceedings of the Workshop on Automatic Summarization, ANLP-NAACL2000*. Seattle, WA.

Mani, Inderjeet, Therese Firmin, David House, Gary Klein, Beth Sundheim, and Lynette Hirschman. 1999a. The TIPSTER Summac Text Summarization Evaluation. In *Proceedings of the 9th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*, 77–85.

Mani, Inderjeet, Thérèse Firmin, and Beth Sundheim. 2002. SUMMAC: A Text Summarization Evaluation. *Natural Language Engineering* .

Mani, Inderjeet, Barbara Gates, and Eric Bloedorn. 1998a. Using Cohesion and Coherence Models for Text Summarization. In *Intelligent Text Summarization Symposium*, 69–76. Standford (CA), USA.

Mani, Inderjeet, Barbara Gates, and Eric Bloedorn. 1999b. Improving summaries by revising them. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics(ACL 99)*, 558–565. Maryland.

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

Mann, William, and Sandra Thompson. 1988. Rhetorical Structure Theory: towards a functional theory of text organization. *Text* 8(3): 243–281.

Marcu, Daniel. 1997. From Discourse Structures to Text Summaries. In *The Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 82–88. Madrid, Spain.

Marcu, Daniel. 1998. To Build Text Summaries of High Quality, Nuclearity is not Sufficient. In *Intelligent Text Summarization*, 1–8. Standford (CA), USA.

Marcu, Daniel. 1999. The automatic construction of large-scale corpora for summarization research. In M. Hearst, Gey. F., and R. Tong, eds., *Proceedings of SIGIR'99. 22nd International Conference on Research and Development in Information Retrieval*, 137–144. University of California, Berkely.

Marcu, Daniel, and Laurie Gerber. 2001. An Inquiry into the Nature of Multidocument Abstracts, Extracts, and Their Evaluation. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*. NAACL, Pittsburgh, PA.

McGirr, Clinton J. 1973. Guidelines for abstracting. *Technical Communication* 25(2): 2–5.

McKeown, Kathleen, Desmond Jordan, and Vassileios Hatzivassiloglou. 1998. Generating patient-specific summaries of on-line literature. In *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06*, 34–43. Standford (CA), USA: The AAAI Press.

McKeown, Kathleen, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards Multidocument Summarization by Reformulation: Progress and Prospects. In *AAAI/IAAI*, 453–460.

McKeown, Kathleen R., and Dragomir R. Radev. 1995. Generating Summaries of Multiple News Articles. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 74–82. Seattle, Washington.

McKeown, Kathleen R., Jacques Robin, and Karen Kukich. 1995. Generating Concise Natural Language Summaries. *Information Processing & Management* 31(5): 702–733.

Michaelson, Herbert B. 1980. *How to Write and Publish Engineering Papers and Reports*. Phoenix, AZ: Oryx Press.

Miike, S., E. Itoh, K. Ono, and K. Sumita. 1994. A Full-text Retrieval System with A Dynamic Abstract Generation Function. In W.B. Croft and C.J. van Rijsbergen, eds., *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 152–161. July 3-6, Dublin, Ireland.

Mikheev, Andrei. 2000. Tagging Sentence Boundaries. In *Proceedings of NAACL 2000*, 264–271.

Minel, Jean-Luc, Sylviane Nugier, and Gérard Piat. 1997. Comment Apprécier la Qualité des Résumés Automatiques de Textes? Les Exemples des Protocoles FAN et MLUCE et leurs Résultats sur SERAPHIN. In *1éres Journées Scientificques et Techniques du Réseau Francophone de l'Ingénierie de la Langue de l'AUPELF-UREF.*, 227–232.

Mitra, Mandar, Amit Singhal, and Chris Buckley. 1997. Automatic Text Summarization by Paragraph Extraction. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, 39–46. Association for Computational Linguistics, Madrid, Spain.

Morris, A., 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.

Mostafa, J., and Wai Lam. 2000. Automatic Classification Using Supervised Learning in a Medical Document Filtering Application. *Information Processing and Management* 36(3): 415–444.

Mostafa, J., S. Mukhopadhyay, Wai Lam, and M. Palakal. 1997. A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation. *ACM Transactions on Information Systems* 15(4): 368–399.

Nakao, Yoshio. 2000. An Algorithm for one-page summarization of a Long Text Based on Thematic Hierarchy Detection. In *Proceedings of the 38th Annual Meeting of the Association for Computation Linguistics*, 302–309.

Nanba, H., and Manabu Okumura. 2000. Producing More Readable Extracts by Revising Them. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, 1071–1075.

Nie, J.Y., Michel Simard, Pierre Isabelle, and R. Durand. 1999. Cross-Language Information Retrieval Based on Parallel Texts and Automatic Mining of Parallel Texts from the Web. In *ACM SIGIR*, 74–81.

Nomoto, Tadashi, and Yuji Matsumoto. 2001. A New Approach to Unsupervised Text Summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*.

Oka, Mamiko, and Yoshihiro Ueda. 2000. Evaluation of Phrase-representation Summarization based on Information Retrieval Task. In *Proceedings of the Workshop on Automatic Summarization, ANLP-NAACL2000*. Seattle, WA.

Okumura, Manabu, H. Mochizuki, and H. Nanba. 1999. Query-biased Summarization Based on Lexical Chaining. In *Proceedings of PACLING'99*, 324–334.

Okurowski, Mary Ellen, Harold Wilson, Joacquin Urbina, Tony Taylor, Ruth Colvin Clark, and Frank Krapcho. 2000. A Text Summarizer in Use: Lessons Learned from Real World Deployment and Evaluation. In *Proceedings of the Workshop on Automatic Summarization, ANLP-NAACL2000*. Seattle, WA.

Ono, K., K. Sumita, and S. Miike. 1994. Abstract Generation Based on Rhetorical Structure Extraction. In *Proceedings of the International Conference on Computational Linguistics*, 344–348. yoto, Japan.

Paice, C.D., and M.P. Oakes. 1999. A Concept-Based Method for Automatic Abstracting. Technical Report Research Report 27, Library and Information Commission.

Paice, Chris. 1989. Automatic Generation and Evaluation of Back-of Book Indexes. In *Prospects for Intelligent Retrieval*.

Paice, Chris. 1990. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management* 26(1): 171–186.

Paice, Chris, and P.A. Jones. 1993. The Identification of Important Concepts in Highly Structured Technical Papers. In R. Korfhage, E. Rasmussen, and P. Willett, eds., *Proc. of the 16th ACM-SIGIR Conference*, 69–78.

Paice, Chris D. 1981. The Automatic Generation of Literary Abtracts: An Approach based on Identification of Self-indicating Phrases. In O.R. Norman, S.E. Robertson, C.J. van Rijsbergen, and P.W. Williams, eds., *Information Retrieval Research*. London: Butterworth.

Palmer, Martha, and Zhibiao Wu. 1995. Verb semantics for english-chinese translation. *Machine Translation Journal* .

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

Pirkola, A. 1998. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *ACM SIGIR*, 55–63.

Pollock, J., and A. Zamora. 1975. Automatic Abstracting Research at Chemical Abstracts Service. *Journal of Chemical Information and Computer Sciences* 15(4).

Preston, Keith, and Sandra Williams. 1994. Managing the information overload. *Physics in Business* .

Radev, Dragomir. 2000. A Common Theory of Information Fusion from Multiple Text Sources, Step One: Cross-document Structure. In *Proceedings, 1st ACL SIGDIAL Workshop on Discourse and Dialogue*. Hong Kong.

Radev, Dragomir, and Weiguo Fan. 2000. Automatic summarization of search engine hit lists. In *Proceedings, ACL Workshop on Recent Advances in NLP and IR*. Hong Kong.

Radev, Dragomir R., Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. 2001a. Interactive, Domain-Independent Identification and Summarization of Topically Related News Articles. In *5th European Conference on Research and Advanced Technology for Digital Libraries*. Darmstadt, Germany.

Radev, Dragomir R., Sasha Blair-Goldensohn, Zhu Zhang, and Revathi Sundara Raghavan. 2001b. NewsInEssence: A System for Domain-Independent, Real-Time News Clustering and Multi-Document Summarization. In *Human Language Technology Conference*. San Diego, CA.

Radev, Dragomir R., Weiguo Fan, and Zhu Zhang. 2001c. WebInEssence: A Personalized Web-Based Multi-Document Summarization and Recommendation System. In *NAACL Workshop on Automatic Summarization*. Pittsburgh, PA.

Radev, Dragomir R., Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *ANLP/NAACL Workshop on Summarization*. Seattle, WA.

Radev, Dragomir R., and Kathleen R. McKeown. 1998. Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics* 24(3): 469–500.

Rath, G., A. Resnick, and R. Savage. 1961. The formation of abstracts by the selection of sentences: Part 1: sentence selection by man and machines. *American Documentation* 12(2): 139–141.

Rau, Lisa F., and R. Brandow. 1993. Domain-Independent Summarization of News. Dagstuhl Seminar, Summarizing Text for Intelligent Communication.

Rau, Lisa F., Paul S. Jacobs, and U. Zernik. 1989. Information Extraction and Text Summarization using Linguistic Knowledge Acquisition. *Information Processing & Management* 25(4): 419–428.

Reiter, Ehud, and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge, U.K.: Cambridge University Press.

Rino, Lucia H.M., and Donia Scott. 1994. Automatic Generation of Draft Summaries: Heuristics for Content Selection. Technical Report ITRI-94-8, Information Technology Research Institute.

Rowley, Jennifer. 1982. *Abstracting and Indexing*. London, UK: Bingley.

Rumelhart, David E. 1975. Notes on a Schema for Stories. In *Language, Thought, and Culture. Advances in the Study of Cognition*. Academic Press, Inc.

Saggion, Horacio. 1999. Using Linguistic Knowledge in Automatic Abstracting. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 596–601. Maryland, USA.

Saggion, Horacio. 2000. Génération automatique de résumés par analyse sélective. Ph.D. thesis, Département d'informatique et de recherche opérationnelle, Faculté des arts et des sciences, Université de Montréa, Montréal, Québec, Canada.

Saggion, Horacio, and Guy Lapalme. 2000a. Concept Identification and Presentation in the Context of Technical Text Summarization. In *Proceedings of the Workshop on Automatic Summarization. ANLP-NAACL2000*. Association for Computational Linguistics, Seattle, WA, USA.

Saggion, Horacio, and Guy Lapalme. 2000b. Selective Analysis for Automatic Abstracting: Evaluating Indicativeness and Acceptability. In *Proceedings of the Computer-Assisted Information Searching on Internet Conference. RIAO'2000*. Paris, France.

Salton, G., and M.J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.

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

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

Salton, Gerard, James Allan, Chris Buckley, and Amit Singhal. 1994. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science* 264: 1421–1426.

Salton, Gerard, James Allan, and Amit Singhal. 1996. Automatic Text Decomposition and Structuring. *Information Processing & Management* 32(2): 127–138.

Saracevic, Tefko. 1975. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science* 26(6): 321–343.

Schamber, Linda, Michael B. Eisenberg, and Michael S. Nilan. 1990. A re-examination of relevance: Toward a dynamic, situational definition. *Information Processing and Management* 26: 755–776.

Shank, R., and R. Abelson. 1977. *Scripts Plans Goals and Understanding*. Lawrence Erlbaum Associates, Publishers.

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

Sherrard, Carol. 1985. The psychology of summary writing 15(3): 247–258.

Siegel, Sidney, and N. John Jr. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences*. Berkeley, CA: McGraw-Hill, 2nd edn.

Silber, H. Gregory, and Kathy McCoy. 2000. Efficient Text Summarization Using Lexical Chains. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI'2000)*.

Simard, Michel, George. F. Foster, and Pierre Isabelle. 1992. Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, 67–81. Montreal, Canada.

Skorochod'ko, E. 1972. Adaptive method of automatic abstracting and indexing. In C. Freiman, ed., *Information Processing 71: Proceedings of the IFIP Congress 71*, 1179–1182. North-Holland Publishing Company.

Somers, H., B. Black, J. Ellman, L. Gilardoni, T. Lager, A. Multari, J. Nivre, and A. Rogers. 1997. Multilingual Generation and Summarization of Job Adverts: The TREE Project. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 269–276.

Sparck-Jones, Karen. 1993a. Discourse modelling for automatic summarising. Technical Report Technical Report No. 290, University of Cambridge, Computer Laboratory.

Sparck-Jones, Karen. 1993b. What might be in a summary. Information Retrieval 93: Von der Modellierung zur Anwendung, 9–26.

Sparck-Jones, Karen. 1997. Summarizing: Where are we now? Where should we go? In I. Mani and M. Maybury, eds., *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain.

Spark Jones, K., and J.R. Galliers. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*. No. 1083 in Lecture Notes in Artificial Intelligence. Springer.

Sperer, Ruth, and Douglas W. Oard. 2000. Structured Translation for Cross-Language Information Retrieval. In *ACM SIGIR*, 120–127.

Tait, John I. 1982. Automatic summarising of English texts. Technical Report Technical Report No. 47, University of Cambridge, Computer Laboratory.

Tait, John I. 1985. Generating summaries using a script based language analyzer. In *Progress in artificial intelligence*.

Tan, Chew Lim, and Makoto Nagao. 1995. Automatic Alignment of Japanese-Chinese Bilingual Texts. *IEICE Transactions On Information and Systems* E78-D(1).

Teufel, Simone. 1998. Meta-Discourse Markers and Problem-Structuring in Scientific Texts. In M. Stede, L. Wanner, and E. Hovy, eds., *Proceedings of the Workshop on Discourse Relations and Discourse Markers, COLING-ACL'98*, 43–49.

Teufel, Simone, and Marc Moens. 1997. Sentence extraction as a classification task. In *The Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain.

Teufel, Simone, and Marc Moens. 1998. Sentence Extraction and Rhetorical Classification for Flexible Abstracts. In *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06*, 16–25. Standford (CA), USA: The AAAI Press.

Teufel, Simone, and Moens Moens. 1999. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In I. Mani and M.T. Maybury, eds., *Advances in Automatic Text Summarization*, 155–171. The MIT Press.

Tombros, Anastasios, Mark Sanderson, and P. Gray. 1998. Advantages of Query Biased Summaries in Information retrieval. In *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06*, 34–43. Standford (CA), USA: The AAAI Press.

Valenza, Robin, Tony Robinson, Marianne Hickey, and Roger Tucker. 1999. Summarisation of spoken audio through information extraction. In *Proceedings of the ESCA workshop: Accessing information in spoken audio*, 111–116.

van Dijk, Teun A. 1979. Recalling and Summarizing Complex Discourse. In W. Burchart and K. Hulker, eds., *Text Processing*.

van Dijk, Teun A. 1988. *News as Discourse*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Waibel, Alex, Michael Bett, and Michael Finke. 1998. Meeting browser: Tracking and summarising meetings. In *Proceedings of the DARPA Broadcast News Workshop*.

Wakao, T., T. Ehara, and K. Shirai. 1998. Text Summarization for Production of Closed-Caption TV Programs in Japanese. *Computer Processing of Oriental Languages* 12(1): 87–97.

Winograd, Peter N. 1984. Strategic difficulties in summarizing texts. *Reading Research Quarterly* 19(4): 404–425.

Wu, Dekai. 1994. Aligning Parallel English Chinese Text Statistically with Lexical Criteria. In *ACL-94*.

Xia, Fei, Martha Palmer, Nianwen Xue, Mary Ellen Okurowski, John Kovarik, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. Athens, Greece.

Yang, Yiming, Jaime G. Carbonell, Ralf D. Brown, and Robert E. Frederking. 1998. Translingual information retrieval: learning from bilingual corpora. *Artificial Intelligence* 103: 323–345.

Young, S. R., and P. J. Hayes. 1985. Automatic classification and summarization of banking telexes. In *Proceedings of the 2nd Conference on Artificial Intelligence Applications (CAIA)*, 402–408. Miami beach, FL.

Zechner, Klaus. 1995. Automatic text abstracting by selecting relevant passages. Master's thesis, Centre for Cognitive Science, University of Edinburgh.

Zechner, Klaus, and Alex Waibel. 2000. Minimizing Word Error Rate in Textual Summaries of Spoken Language. 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 (ANLP-NAACL'2000*, 186–193.

Zhang, Zhu, Sasha Blair-Goldensohn, and Dragomir Radev. 2002. Towards CST-Enhanced Summarization. AAAI 2002.