The Elements of Automatic Summarization

by

Daniel Jacob Gillick

A dissertation submitted in partial satisfaction
of the requirements for the degree of

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Nelson Morgan, Chair
Professor Daniel Klein
Professor Thomas Griffiths

Spring 2011

Abstract

The Elements of Automatic Summarization
by
Daniel Jacob Gillick
Doctor of Philosophy in Computer Science
University of California, Berkeley
Professor Nelson Morgan, Chair

This thesis is about automatic summarization, with experimental results on multi-document news topics: how to choose a series of sentences that best represents a collection of articles about one topic. I describe prior work and my own improvements on each component of a summarization system, including preprocessing, sentence valuation, sentence selection and compression, sentence ordering, and evaluation of summaries. The centerpiece of this work is an objective function for summarization that I call "maximum coverage". The intuition is that a good summary covers as many possible important facts or concepts in the original documents. It turns out that this objective, while computationally intractable in general, can be solved efficiently for medium-sized problems and has reasonably good fast approximate solutions. Most importantly, the use of an objective function marks a departure from previous algorithmic approaches to summarization.

Acknowledgements

Getting a Ph.D. is hard. Not really hard in the day-to-day sense, but more because I spent a lot of time working on a few small problems, and at the end of six years, I have only made a few small contributions to the world. And here it is, my final attempt to communicate those contributions, typed out over unusual flavors of tea inside Oakland's cafes while the sun drifts past. Snow piled up on the East Coast, Egypt shed its dictator, Libya revolted, a giant earthquake shook Japan. Few people will be aware of my work in this field, and fewer still will read this thesis. So to those who are reading, and to those who helped support me while I computed, wrote, and just sat thinking, thank you!

To: My parents. Not because they are particularly in need of summaries.

Contents

Chapter 1

Introduction

1.1 Who needs summaries?

In the year 77 of the Common Era, Pliny the Elder published his magnum opus, *Naturalis Historia*, an encyclopedic account in 36 chapters of everything he knew. Pliny, a Roman lawyer, army officer, local governor, author, naturalist, and philosopher, had a voracious appetite for knowledge, which he hoped to impart to his friend, the emperor Titus as he set about the task of ruling the unwieldy Roman Empire. The work was groundbreaking, a model for the modern encyclopedia in terms of breadth, citation, and indexing. It covered mathematics, the sciences, agriculture, medicine, and the arts. And, just in case Titus found himself short on reading time, Pliny wrote summaries of each section: the first known abstracts.

Since Pliny's time, the amount of published material has grown rather dramatically. The wisdom implied by his obsessive scholarship ("there is no book so bad that some good could not be got out of it") just sounds foolish in the era of Amazon and Wikipedia. As a result, the value of summaries has likewise increased, even for ordinary citizens.

Nowadays, we are so surrounded by summaries of information that we often take them for granted. Imagine a newspaper without headlines! Books and movies are described in blurbs and reviews, scholarly articles begin with abstracts, and search results are summarized with a few snippets from each page. Summaries are of great potential value to journalists, lawyers, students, CEOs, and casual browsers of the Internet[1].

These days, nearly all summaries we are likely to encounter are written by people. They are typically *abstracts*, distillations of the original written in new language, as opposed to *extracts*, patchworks of text selected from the source (though empirical

[1]Take the recent Internet meme: *TL;DR*, which stands for *Too Long; Didn't Read*.

Figure 1.1: A 12th century version of Pliny the Elder's *Naturalis Historia*.

studies have shown there is more copying than you might expect when the New York Times archivists summarize articles, for example.). Still, writing summaries is something of an art: Experts have trouble explaining exactly what steps they follow in absorbing and reformulating information. Thus, while the prospect of automatic summarization is appealing, it is not surprising that there remains a considerable gap between human and machine-written summaries.

1.2 On the prospect of automation

Research in automatic summarization began in the 1950s. Hans Peter Luhn, a computer scientist at IBM, perhaps best known for developing the algorithm used for validating credit card numbers, published a paper in the 1958 edition of the IBM Journal called *The Automatic Creation of Literature Abstracts* [Luhn, 1958]. Remarkably, Luhn took a mathematical approach to the problem which was largely abandoned and then rediscovered in the last 15 years as statistical techniques have rejuvenated Artificial Intelligence research:

Statistical information derived from word frequency and distribution is

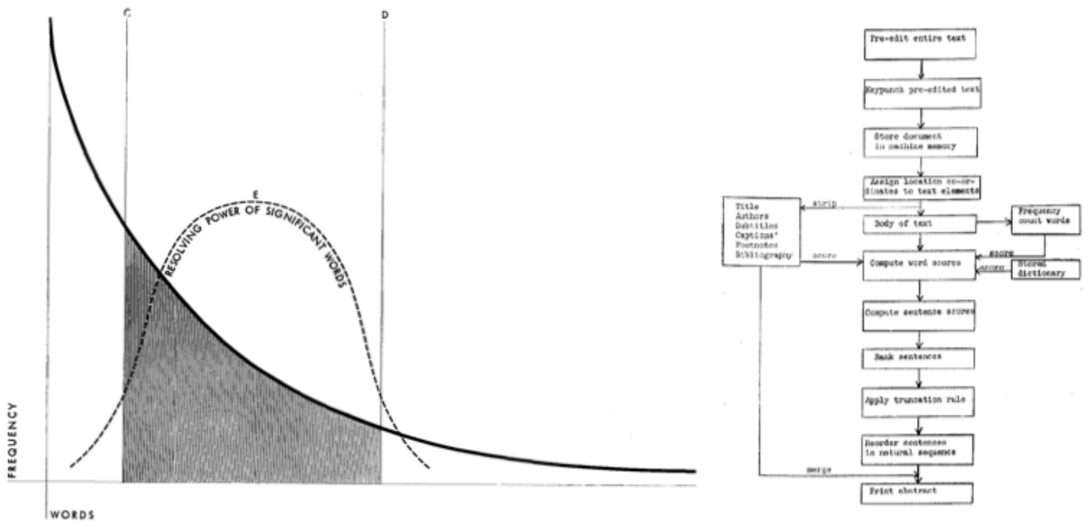


Figure 1.2: Luhn began thinking about statistical properties of documents (Left); Edmundson's system diagram contains many components of a modern summarizer (Right).

> used by the machine to compute a relative measure of significance, first for individual words and then for sentences. Sentences scoring highest in significance are extracted and printed out to become the "auto-abstract."

Luhn had to transcribe each document he wished to summarize onto a punch card, which made things painfully slow, but his approach to summarization is truly foundational. Surprisingly little has changed in today's state-of-the-art systems. And, while some have taken a few small steps towards language generation, extractive summarization is still the focus of most research. H. P. Edmundson's description in 1964 [Edmundson, 1964] is still apt today:

> Present systems of automatic abstracting are capable of producing nothing more than extracts of documents, i.e. a selection of certain sentences of a document. This is not to say, however, that future automatic abstracting systems cannot be conceived in which the computer generates its own sentences by means of a suitable generative grammar program. Theoretically there is no linguistic or mechanical reason why such a system could not be designed and operated [...] Such a system, however, is apt to be costly both in time and money.

What has changed is the context in which we understand the problem. Most publications are available in what Luhn would have called *machine-readable form*, and are readily accessible online. Computing power and accessibility are no longer primary limitations. Practical issues associated with summarization include:

Preprocessing: Many segments of raw text are not suitable for extraction. Tables, quotes, bylines, and all kinds of formatting markup are unwelcome in a summary, and should be removed before analysis begins. This problem is difficult, and often overlooked, but poor preprocessing can eclipse improvements in other aspects of summarization.

Sentence Segmentation: Sentences are a natural unit for extraction, but identifying sentence boundaries is not trivial because periods are also used to denote abbreviations. Segmentation mistakes are costly because a fragmented sentence can ruin a summary.

Valuation: To choose which sentences to include in a summary, we need some way of measuring the value of each sentence, or more generally, selection unit. Luhn proposed inferring the value of a sentence from the number of times the words it contains appear in the original input. Other features of sentences, like their position or the presence of various key phrases like *in conclusion*, have also been studied.

Selection: Luhn's system simply selected the sentences with the largest inferred values. But this neglects redundancy in the relevant sentences. Thus, selection is a constrained optimization problem: maximize relevant content subject to a summary length constraint.

Ordering: The ordering of a set of sentences can dramatically affect the meaning and readability of the summary. While a proper treatment of sentence ordering ought to be more holistic, that is, considered jointly with selection, little work in this area has had much impact.

Evaluation: Measuring and understanding the differences between different summarization systems is crucial to advancing the field. Since manual evaluation by experts is slow and costly, a variety of automatic and semi-automatic alternatives are used. The somewhat standard, though problematic ROUGE metric measures word overlap between machine-generated summaries and a set of human-written abstracts.

Significant research has been devoted to most of these areas, and the next chapters will survey some key results. But often, results are hard to compare. Because most researchers build their own systems from the ground up, without standardized components, it is hard to tell if an improvement is due to a better method for selecting sentences or simply better preprocessing, for example. One motivation for this work is to provide some simple but state-of-the-art components to facilitate further development and comparison.

There are many varieties of summarization. Ongoing research investigates summarization of email, legal proceedings, customer reviews, search results, meetings, and videos. Many of the standard techniques, however, have their origins in document summarization, where the goal is to convey the most important information from a set of documents within a length constraint using natural language. The focus of this thesis is narrow: summarization of multiple news documents, but the ideas are more broadly applicable.

1.3 A note on computers and language

Initially, the prospect of summarizing automatically sounds either impossible or like magic. Considering more carefully how extraction actually works, usually by counting simple features of the words in the documents, can be something of a disappointment. Far from magic, this application of Artificial Intelligence seems unrelated to our intuitive sense of what intelligence is. The computer does not comprehend any meaning; it doesn't even have a framework for representing semantics. All told, we are a long way from creating intelligence of the sort that goes on in the human brain.

So rather than approach problems like summarization by trying to emulate people, we take a far less ambitious tack. Instead, we make simplifying assumptions about what a summary ought to be, and then use the computer for what it does best: calculating quickly. That is, rather than employ a series of human-like steps (pick a "topic sentence", find a series of "supporting sentences", etc.), we use the computer to search for the summary that maximizes the value of a made-up formula (find the set of sentences that together contain the most frequent words in the documents).

This transition, from procedural to statistical thinking, has been amazingly successful in other related fields. Machine translation and speech recognition improved tremendously when researchers stopped trying to write software to imitate people and started fitting statistical models from reams of data. Summarization research has lagged a bit behind in this respect, perhaps because it is hard to think of simple but reasonable models for how a few thousand sentences should be reduced to three or four. Still, the primary philosophical contribution of this work is to help promote the transition to statistical approaches for summarization.

1.4 What to expect

As I write, I am keeping in mind a particular reader. My reader is like me, but just beginning to choose a research topic. Like most areas, there are a great many summarization papers and ideas floating around, but little top-down organization. I am trying to write the thesis that would have been most useful for me to read three years ago, focusing in particular on the work I've done since then. I will try to keep things simple and clear, and avoid making assumptions about what my reader knows and does not know.

The next chapter describes the data used in my experiments and the variety of evaluation methods, both manual and automatic; it also provides some general statistics that highlight differences between summaries and full documents. Chapter 3 discusses preprocessing issues, including the sentence boundary detection problem. Chapter 4 steps through a series of advances in sentence selection methods, highlighting the advantages of an objective function for summarization over procedural approaches. Chapter 5 gives a deeper analysis of the maximum coverage objective proposed in chapter 4, and suggests an expansion for solving selection and sentence compression jointly. Chapter 6 discusses the often ignored problem of sentence ordering, suggesting a simple alternative to the difficult task of finding the the best way to order a small set of sentences. Chapter 7 addresses evaluation. It describes a simpler alternative to ROUGE with more desirable properties, and discusses the results of an evaluation by non-expert annotators. Chapter 8 suggests future directions for summarization.

Chapter 2

News Data

For the sake of consistency and simplicity, all experiments and discussion that follow involve a subset of a dataset called the English Gigaword Corpus (3rd edition) [Graff *et al.*, 2007]. The corpus consists of millions of newswire articles published by six different sources between 1994 and 2006.

2.1 Anatomy of a summarization problem

The summarization problems presented here are the official sets distributed as part of a competition organized annually since 2001 by the National Institute of Standards and Technology (NIST). Each problem was created by an expert who assembled a group of related documents and wrote a "topic" and "narrative description" to help focus the summary. The 2006 and 2007 sets asked for 250 word summaries[1] from groups of 25 documents. In the hopes of nudging research away from pure sentence extraction, the 2008 and 2009 sets asked for 100 word summaries of 10 documents[2]. Each year's problem set includes around 50 such problems.

To facilitate evaluation, a total of four summaries are written by experts for each problem. Notably, these are abstracts rather than extracts. The same experts run a comprehensive evaluation of all the summaries submitted by participating teams. Figure 2.1 shows two sample summaries generated automatically and one written by an expert, along with evaluation results from NIST's 2008 Text Analysis Conference.

[1]This is a hard cap on the number of space-separated words. Other summarization problems employ different sorts of limits, but here we assume that the summary will be truncated after the word limit is reached.

[2]It didn't work. Most submitted systems remain extractive.

Topic:	Airbus A380
Narrative:	Describe developments in the production and launch of the Airbus A380.

Reference

The European Airbus A380 flew its maiden test flight from France 10 years after design development started. The A380 super-jumbo passenger jet surpasses the Boeing 747 and breaks their monopoly. Airlines worldwide have placed orders but airports may need modification to accommodate the weight and width of the A380. U.S. airlines have not placed an order. Airbus has fallen behind in production and a backlog of orders has developed. Airbus must sell at least 250 planes to break even financially. The A380 is overweight and modifications to meet the weight requirements impacted the budget. Additional test flights are planned.

Summary A

European airplane maker Airbus "is likely to discuss before the end of the year" a possible increase in production capacity of its new super-jumbo A380 aircraft, Airbus' production chief Gustav Humbert said in a magazine interview released Tuesday. The superjumbo Airbus A380, the world's largest commercial airliner, took off Wednesday into cloudy skies over southwestern France for its second test flight. This second test flight is part of a program of tests that will intensify from here on out, and we're not going to make statements on each phase of the program," Airbus said.

Summary B

Airbus unveiled the world's biggest passenger jet at a spectacular sound-and-light ceremony in Toulouse. And even though no US airline has ordered the European jet, aviation authorities are getting Los Angeles San Francisco and Miami airports ready for the A-380 passenger flights in 2006. Airbus has 139 firm A380 orders from 13 airlines and freight companies, worth $39 billion before any discounts on the plane's $280 million list price. Airbus sees global demand for 1,250 A380-size behemoths to shuttle passengers between the world's largest airports, which serve as connecting hubs for flights to less busy destinations.

System	OQ	LQ	Pyramid	ROUGE-2
Reference	5	4.5	0.550	0.065
System A	1	3.0	0.000	0.026
System B	4	4.0	0.276	0.048

Figure 2.1: A problem from the 2008 evaluation. One reference summary and two system summaries are shown along with evaluation results. CR is the Content Responsiveness as judged by an expert; LQ is the average of five expert-judged linguistic quality scores. Both are on a 1 (very poor) to 5 (very good) scale. $Pyramid$ measures overlap between the important facts in the summary and the set of references. $ROUGE$ measures overlap in n-grams; $ROUGE\text{-}2$ uses $n = 2$.

2.2 Evaluation

Compared with other natural language processing research areas, summarization is a particularly vague and subjective task. What makes a good summary may vary greatly depending on the circumstances and the intended audience. So while evaluation is essential to the progression of the field, it remains a thorny issue.

As it turns out, evaluating individual summaries is considerably more difficult than evaluating summarization systems. This is because averaging together scores for summaries produced by a single system can eventually reduce variability enough to allow for meaningful comparison of systems. Luckily, we are usually interested in comparing systems rather than summaries, to see whether a system-level change improves average results.

2.2.1 Manual Evaluation

Traditional evaluation of summaries involves human judgments in categories like *content* and *grammaticality*. The annotator is instructed to read the original documents and then score summaries without respect to any particular task or goal. This sort of evaluation is *intrinsic*: it attempts to measure quality directly. *Extrinsic* methods, by contrast, measure performance on a particular task, like timed question answering.

The annual competitions adjusted their criteria somewhat from year to year, but the general setup remained fairly consistent. Scores range from 1 (very poor) to 5 (very good), but changed to a 10-point scale in 2009. Here are the instructions given to the annotators in 2006 and 2007:

Content Responsiveness: Responsiveness should be measured primarily in terms of the amount of information in the summary that actually helps to satisfy the information need expressed in the topic statement. The linguistic quality of the summary should play only an indirect role in your judgment, insofar as poor linguistic quality interferes with the expression of information and reduces the amount of information that is conveyed.

Grammaticality: The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

Non-redundancy: There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.

Referential Clarity: It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

Focus: The summary should have a focus; sentences should only contain information that is related to the rest of the summary.

Structure and Coherence: The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

I will often show *Linguistic Quality* scores, averages across the five grammatical categories, and *Overall Quality* scores, a new category used in the 2008 and 2009 evaluations in place of content responsiveness. One set of instructions suggested that overall quality should be related to the relative dollar value of a summary.

2.2.2 Automatic Evaluation

Manual evaluation is slow and costly[3]. To facilitate comparison between systems, more immediate feedback is important. ROUGE, or Recall-Oriented Understudy for Gisting Evaluation [Lin, 2004], is the awkwardly named adaptation of Machine Translation's BLEU scoring [Papineni *et al.*, 2002] to summarization. ROUGE-n is roughly a measure of the n-gram overlap between a summary and the set of reference summaries, where the value of each n-gram is the number of references in which it appears. It is recall-oriented because the summary length constraint implicitly penalizes the inclusion of irrelevant words. The ROUGE toolkit includes a variety of options (ignore stopwords[4]? perform word stemming before comparing?) and can use word combinations more complex then n-grams: ROUGE-SU4, for example, counts unigrams and pairs of words separated by up to four intervening words.

ROUGE is not good at predicting the manually evaluated quality of a particular summary, but when ROUGE scores are averaged over many problems, the correlation is usually good. Figure 2.2 shows the relationship between ROUGE and manually evaluated quality in the 2008 evaluation. To achieve 90% correlation for this data, ROUGE-2 requires around 30 problems, and with nearly 100 problems, the correlation improves to almost 95%. The figure also suggests that with fewer than 15 topics, ROUGE-1 is actually more reliable than higher order n-grams. In general, while higher order n-grams are more meaningful, more predictive, they are rarer, so their absence in a particular summary might not be significant.

ROUGE certainly permits rapid system development, but it is an easy target for criticism. Given the references, it is a simple matter to create completely nonsensical

[3]It is also highly subjective; one additional advantage of ROUGE is that scores are readily comparable across evaluations, whereas different annotators are likely to have different personal score ranges.

[4]Stopword lists include common function (as opposed to content) words like articles, prepositions, and pronouns)

but high-ROUGE-scoring summaries. And, as with any simple automatic metric, the more it is used to optimize machine learning methods, the less useful it becomes in evaluation. While it is still used during development, good research papers cite results of manual comparisons to support their findings.

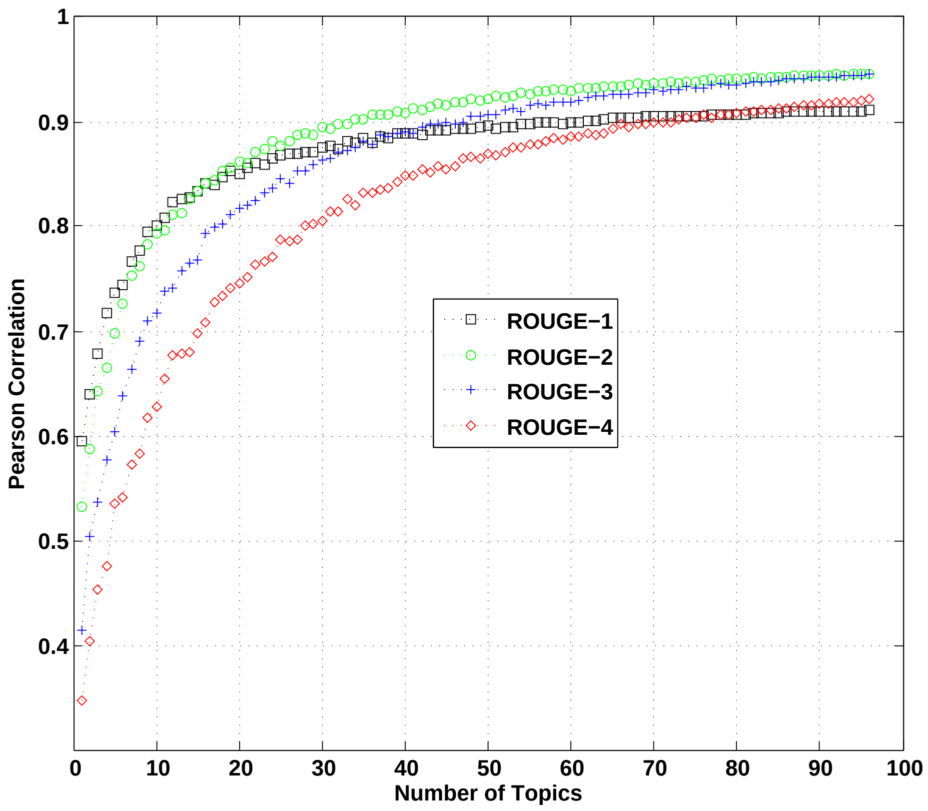


Figure 2.2: Correlation between ROUGE-n and human-evaluated summary quality when scores are averaged over different numbers of topics. (2008 data).

2.2.3 Semi-automatic evaluation

The primary shortcoming of ROUGE is that it treats word n-grams as if they were facts. Nenkova's Pyramid method for evaluation [Nenkova and Passonneau, 2004] is an attempt to bridge the gap between fully manual evaluation and ROUGE. Rather than rely on n-grams as a unit of information, *Semantic Content Units* are manually identified in the reference summaries (phrases, sentence fragments, etc.) so that a system summary can be scored based on the relevant facts it contains. Pyramid shows considerably stronger correlation than ROUGE with manual responsiveness, and remarkably, a simple linear regression over average Pyramid scores and average linguistic quality scores can almost exactly predict overall responsiveness (correlation is 99%). Note that this regression is at the system level: very little work has been done on evaluating individual summaries.

The Pyramid method is not particularly automatic: human annotators are needed to identify facts in the references and match these facts in new summaries. Still, it is an attempt to begin mechanizing the identification of important content, separate from linguistic quality, in summaries. And, Pyramid scores reasonably represent the gap between human summaries and automatically generated summaries: While human ROUGE scores just about match the best systems', human Pyramid scores are about twice as large as the best systems'.

2.3 Summaries versus documents

Before diving into a discussion of the different components involved in a summarization system, it is helpful to outline some of the differences between the human-written summaries and the original documents in our data.

To give some intuition about which words translate from documents into summaries, Table 2.1 shows the estimated probability that a word appears in a summary given that it appeared somewhere in the source documents, for some sample words. Note that for this dataset, the word token-level compression rate, from documents to summaries is 2.8%; the word type-level compression rate is 19.4%: the words in 10 related documents, of course, are much more redundant than the text in a single summary.

Topical words like *Russia* or *FBI* carry enough meaning to merit use in a summary, but relative dates, question words, and conversational pronouns (*I*, *you*), usually don't. That articles, conjunctions, and common prepositions appear in both documents and summaries is of little consequence as they are required for proper grammar in any text.

Table 2.2 highlights words that would likely be useful to a classifier in distinguishing summaries from documents using the piecewise (per word) symmetric Kullback-Leibler Distance (KLD) between the distribution over words in the documents $P_D(\cdot)$ and words in the summaries $P_S(\cdot)$:

$$KLD_w = P_D(w) \log \frac{P_D(w)}{P_S(w)} + P_S(w) \log \frac{P_S(w)}{P_D(w)} \tag{2.1}$$

Over 6% of document words are *the*, but only 1.4% of summary words, an indication that human abstractors use different syntactic forms. This claim is reinforced by the distributional differences in the words *of*, *to*, *a*, and *that*. First- and second-person personal pronouns *we*, *our*, and *I* almost never appear in summaries where facts

Word	$\hat{P}(w \in S \mid w \in D)$	Word	$\hat{P}(w \in S \mid w \in D)$	Word	$\hat{P}(w \in S \mid w \in D)$
the	0.9766	russia	0.5167	think	0.0144
and	0.9375	china	0.4118	ago	0.0132
to	0.9271	prison	0.3889	i	0.0130
of	0.8464	fbi	0.3750	thursday	0.0074
in	0.8411	arrest	0.3667	wednesday	0.0071
a	0.8151	indian	0.3542	you	0.0047
for	0.7057	france	0.3500	u.s.	0.0042
on	0.5443	drug	0.3421	come	0.0038
was	0.4427	nuclear	0.3409	monday	0.0035
with	0.4141	judge	0.3365	what	0.0033

Table 2.1: The estimated probability that a word appears in a 100 word summary given that it appeared in at least one of the 10 source documents: words with the highest probability (left column); content words with the highest probability (middle column); words with the lowest probability (right column). Only words appearing in at least 10 different problems are shown (2008 data).

dominate opinions. Months and years (absolute dates) are common in summaries which often give chronologies, while relative dates (Tuesday) lose meaning without context. Also, note that *said* appears nearly eight times more frequently in documents because summaries typically disregard attribution phrases so common in news documents (*"..., sources said"*).

Table 2.3 gives distributional statistics for part-of-speech tags. Overall, nouns are used more in documents (with the exception of plural proper nouns) while verbs are used more in summaries. These differences are quite striking: NNs appear with a third their ordinary rate in summaries; even proper nouns, which we might expect to be particularly relevant are less than half as frequent in summaries. The various verb varieties are roughly twice as prevalent in summaries, which, these results suggest, are about *actions*. Summaries also contain more comparative adjectives and adverbs, and fewer prepositions and determiners.

What are we to make of these differences? How are words and phrases kept or removed in summary creation? One way to understand summarization from a structural perspective is through the Machine Translation concept of *alignment*[5]. The idea, as it applies to translation, is that a pair of parallel sentences in two different languages share a hidden alignment, a mapping of words or phrases or syntactic subtrees that shows how meaning is transferred. Armed with an alignment model trained from a corpus of parallel sentences, we can translate a new sentence by decoding it: find the sequence of words in the other language that maximizes the alignment

[5]See Kevin Knight's excellent tutorial [Knight, 1999].

Word	KLD	$\hat{P}_D(w)$	$\hat{P}_S(w)$	Word	KLD	$\hat{P}_D(w)$	$\hat{P}_S(w)$
the	0.0719	0.0623	0.014069	2005	0.0112	0.000082	0.0031
we	0.0227	0.0022	$< 10^{-7}$	u.s.	0.0053	0.000799	0.0040
said	0.0192	0.0109	0.001426	2004	0.0018	0.000332	0.0015
of	0.0131	0.0279	0.012193	january	0.0017	0.000171	0.0018
to	0.0107	0.0278	0.013356	continue	0.0016	0.000459	0.0017
a	0.0097	0.0247	0.011743	include	0.0015	0.000991	0.0026
our	0.0071	0.0007	$< 10^{-7}$	investigation	0.0015	0.001023	0.0026
tuesday	0.0066	0.0007	$< 10^{-7}$	japan	0.0013	0.000087	0.0007
that	0.0061	0.0131	0.005778	2006	0.0013	0.000129	0.0008
i	0.0060	0.0023	0.000150	april	0.0012	0.000197	0.0009

Table 2.2: The piecewise Kullback-Leibler Distance (KLD) is shown between the empirical distributions over words in the documents $P_D(\cdot)$ and words in the summaries $P_S(\cdot)$: highest KLD words appearing more frequently in the documents (left); highest KLD words appearing more frequently in the summaries (right). (2008 data).

probability given by the model.

Daume and Marcu try applying the alignment idea to abstracts and source documents [Daumé III and Marcu, 2005]. Human annotators create a set of roughly 2000 phrase-aligned pairs with reasonably good inter-annotator agreement, that they use to fit a model (that ends up being fairly complex) giving better precision and recall than more heuristic baselines[6]. While this is an intriguing avenue of research, leading away from extraction, I will not pursue it here. However, it would be nice to build a purely abstractive summarization system that combined the alignment or *translation* model hypotheses with a language model (as in Machine Translation) to produce summaries. They would be somewhat nonsensical because both the alignment model and standard n-gram language models are too weak for actual language generation. But at least the summaries would be amusing.

[6]See also Jing and McKeown's work on mapping summary text to source text [Jing and McKeown, 1999]: In particular, a study of single news document summaries shows that nearly 80% of abstract sentences are derived from cutting and pasting phrases from the source.

Tag	Examples	$\hat{P}_D(t)$	$\hat{P}_S(t)$
NN	government year state time court	0.1323	0.0486
IN	of in for on that	0.1102	0.0486
NNP	bush abramoff house senate	0.1020	0.0481
DT	the a an this that	0.0900	0.0485
"	"	0.0220	0.0154
JJ	other last new first many	0.0593	0.0486
NNS	people years officials days members	0.0573	0.0483
,	,	0.0532	0.0453
PRP	he it they we i	0.0261	0.0258
$	$	0.0011	0.0063
JJR	more less better larger smaller	0.0027	0.0111
()	()	0.0026	0.0096
NNPS	states democrats republicans americans	0.0027	0.0091
RBR	more earlier less longer better	0.0012	0.0038
JJS	most least largest best latest	0.0019	0.0059
POS	's	0.0092	0.0267
EX	there	0.0012	0.0033
PRP$	his their its her our	0.0113	0.0282
RP	up out down off on	0.0031	0.0077
MD	will would could can should	0.0096	0.0238
:	: ; ... - –	0.0019	0.0048
RBS	most best	0.0005	0.0011
WDT	that which whatever what	0.0048	0.0113
VBG	being according including saying going	0.0181	0.0409
VBP	are have do say 're	0.0132	0.0275
CD	one two million 000 three	0.0212	0.0424
WRB	when where how why wherever	0.0032	0.0065
TO	to	0.0242	0.0450
CC	and but or nor	0.0254	0.0468
VBN	been made known used expected	0.0252	0.0454
VBZ	is has 's says does	0.0187	0.0295
VB	be have make take get	0.0286	0.0444
WP	who what whom whoever whole	0.0043	0.0065
RB	not also n't now only	0.0295	0.0411
.	. ? !	0.0373	0.0486
PDT	all such half	0.0004	0.0005
VBD	said was had were did	0.0442	0.0445

Table 2.3: Comparison of the empirical distribution of part-of-speech tags in documents $\hat{P}_D(t)$ and summaries $\hat{P}_S(t)$. The most common example words are shown for each tag type. 2008 data (results are nearly identical with 2009 data).

Chapter 3

Preprocessing: Preparing Text

Text is widely available, but there are few consistent standards, so different sources look quite different. Still, the preparation of raw text for summarization, or other natural language processing tasks, is often neglected by researchers as mundane. This chapter discusses formatting, removal of unwanted phrases (a kind of compression), and sentence segmentation, in each case showing why the problem deserves more serious consideration.

3.1 Cleaning up formatted text

Each publication has its own standards, which evolve constantly. The New York Times, Xinhua, and Associated Press documents in our dataset, for example, have tables, bulleted lists, bylines, location headers, image captions, and various kinds of contact information all lurking in the text, often indicated by different markers. Table 3.1 gives examples of headers. Removing such content-less segments is important for two reasons: It improves the readability of extracted sentences and it clears space for more real content.

As a result, each summarization system includes code for handling formatted text, usually a set of carefully designed, highly specialized regular expressions. Anecdotal reports suggest that some teams devote hundreds of hours to this process. While automated text cleaning would be valuable, there is no research, to my knowledge, in this area, at least in the context of summarization. One way to address cleaning a bit more systematically is to segment text into sentences and then prune those rejected by a parser or without a sufficient number of dictionary words. But such methods are too coarse: If there are oddly formatted bylines attached to the first sentence of each document, these sentences, though otherwise useful, would be discarded.

While I have not implemented any systematic solution (I use a limited set of

Original	**Cleaned**
Vatican-pope-US-women,newseries WASHINGTON Pope John Paul II is credited for some advances by women in the Catholic Church.	Pope John Paul II is credited for some advances by women in the Catholic Church.
New Orleans – A thoroughfare in New Orleans that ends where the water begins has become a launching point for rescue workers.	A thoroughfare in New Orleans that ends where the water begins has become a launching point for rescue workers.

Table 3.1: Examples of undesirable, source-specific formatting that is removed from the first sentences of news documents using regular expressions (2008 data).

regular expressions), one way to approach the problem is to assume two underlying states: good and bad. Usable text is generated by the good state and unusable text is generated by the bad state. Under the Hidden Markov Model framework, we would learn *emission* probability distributions $P(text|good)$ and $P(text|bad)$ based on features of the text like letter n-grams, and *transition* distributions over sequences of underlying states. Labeled data for supervised learning can come from the output of a rule based system. The derived labels for new data can be inferred from a Viterbi decoding, and sections marked as bad can be excised. Most likely, some additional constraints on keeping sentences intact would be necessary, and parameters might need adjusting for new domains. Still, this would be a step towards better automation of a painful part of summarization.

3.2 Sentence compression as preprocessing

Beyond formatting issues, there are segments of clean text that have no bearing on the summary. Removing unnecessary parts of sentences while maintaining meaning and grammar is called sentence compression. The idea was pioneered by Knight and Marcu [2000], who trained a noisy-channel style system[1] to remove subtrees from the parse tree of an original sentence. Subsequent work improved the readability of the compressed sentences [McDonald, 2006], but no results have shown significant improvements in overall summary quality, despite a number of attempts [Lin, 2003; Nenkova, 2008; Martins and Smith, 2009; Gillick and Favre, 2009].

Why should this be? One problem is that compression invariably introduces some semantically or grammatically bad sentences, which work against gains in information content. Perhaps the real problem is a bit more subtle. Most systems create

[1]As in speech recognition and machine translation, a combination of an acoustic or translation model and a language model is used.

compressed sentence alternatives and allow the inference algorithm to choose among them. But since the words and phrases we want to remove (days of the week, attribution clauses, etc.) are common, they are likely to receive unsuitably high weight during valuation. Either we should learn to compress and extract sentences jointly, or compression should precede extraction.

To demonstrate the potential of preprocessing compression, I wrote a few regular expressions to remove some days of the week and attribution clauses. The regular expression for days of the week is shown here (note: this is not simple!), which is designed to have high recall[2]—virtually no syntactic or semantic issues arise as a result of the deletions:

```
s = re.sub(r'(^| |, )(on|last|this|next|late)
        (monday|tuesday|wednesday|thursday|friday|saturday|sunday)
        ( morning| afternoon| evening| night)?
        ($| |,|\.)', r'\5', s)
```

Table 3.2 gives some sense for how this sort of compression can improve performance. The differences in ROUGE score are significant and more improvement seems possible if more rules are added, or if the rules are replaced by a statistical system.

	Compression	ROUGE-2	LQ	Words per Sent.
2008	None	0.108	4.95	24.8
	Rules	0.114	N.A.	24.0
2009	None	0.110	N.A.	26.3
	Rules	0.114	5.40	25.4

Table 3.2: A summary of the performance gap created by more extensive text cleaning. The Linguistic Quality (LQ) scores are not directly comparable since not all of these systems were submitted for evaluation. The 2008 LQ score is multiplied by two here since the scale was changed from 1-5 to 1-10 in 2009. The differences in ROUGE-2 and the average number of words per summary sentence are significant at 95% confidence.

3.3 Sentence segmentation

The sentence segmentation problem—the disambiguation of periods that arises because periods signal abbreviations as well as sentence boundaries—is mostly disregarded because a few rules treat most examples effectively. However, even the

[2]In the 2008 data, 8.7% of input sentences contain at least one of these days of the week; after application of this rule, 5.7% have at least one.

strongest rule-based system [Aberdeen *et al.*, 1995], highly specialized and labor intensive to create, has an error rate of 1% on a standard corpus of Wall Street Journal text [Marcus *et al.*, 1993]. The unsupervised *Punkt* system [Kiss and Strunk, 2006] is widely used (1.65% error rate on the Wall Street Journal corpus; 3% error rate with the included model[3]). Similar error rates (close to 1%) were reported with supervised systems [Palmer and Hearst, 1997; Reynar and Ratnaparkhi, 1997].

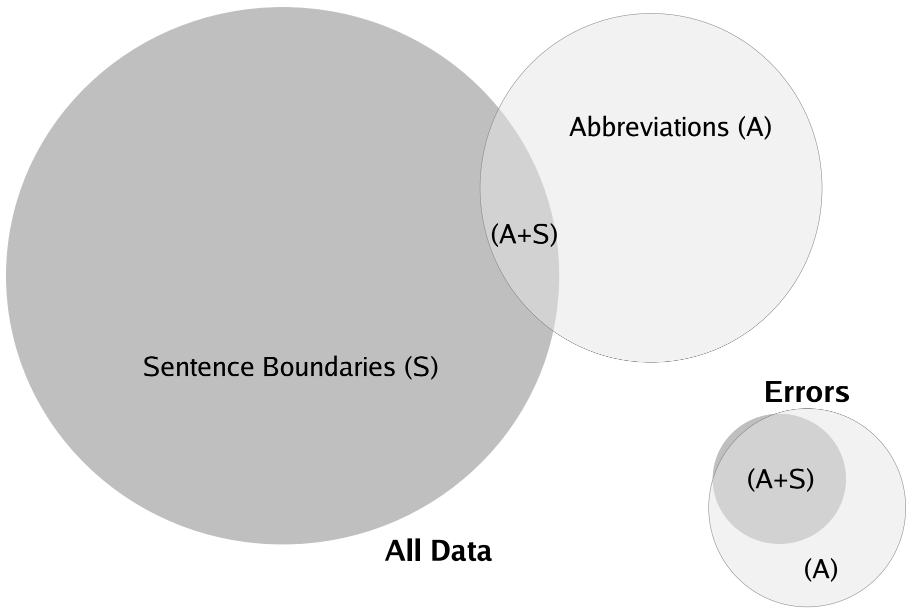


Figure 3.1: The overlapping label space of the Wall Street Journal test data: sentence boundaries (76%), abbreviations (26%), intersection (2%). The distribution of errors given by the classifier is shown as well (not to scale with all data).

All these systems either employ a hard-coded list of abbreviations or seek to compile one automatically. This appears to be a mistake. As Figure 3.1 shows, most of the difficult cases involve abbreviations that can also end sentences (*he was born in the U.S.*). Based on this observation, I built a new system for classifying sentence boundaries that achieves an error rate of 0.25% on the Wall Street Journal Corpus [Gillick, 2009].

The success of this system is partly due to a rich feature set. Each example takes the general form "$L.\ R$", where L is the context on the left side of the period in question, and R is the context on the right (only one word token of context on each side is used). We are interested in the probability of the binary sentence boundary class s, conditional on its context: $P(s|"L.\ R")$. Features used in supervised learning

[3] *Punkt* ships with the Natural Language Processing Toolkit [Loper and Bird, 2002].

are extracted from "*L. R*", described in Table 3.3. Each individual feature is binary, so the actual feature vector for an example is very large but very sparse.

#	Feature Description	Error
1	Left context	1.88%
2	Right context	9.36%
3	Length of left context	9.12%
4	Capitalization of right context	12.56%
5	Log count of left context without following period in the training set	12.14%
6	Log count of right context appearing lowercased in the training set	18.79%
7	Left context and right context	10.01%
8	Left context and capitalization of right context	7.54%
{1,2}		0.77%
{1,2,3,4}		0.36%
{1,2,3,4,5,6}		0.32%
{1,2,3,4,5,6,7,8}		0.25%

Table 3.3: The eight feature classes. All individual features are binary. SVM classification results shown (linear kernel).

Feature classes 1 and 2 together give surprisingly good performance. Adding feature classes 3 and 4 better than cuts the remaining errors in half. The length of the left token serves as a proxy for the abbreviation class (mean abbreviation length is 2.6, compared to 6.1 for non-abbreviation sentence enders). The capitalization of the right token is a proxy for a sentence starter. Every new sentence that starts with a word (as opposed to a number or punctuation) is capitalized, but 70% of words following abbreviations are also, so this feature is mostly valuable in combination.

While the training set includes nearly a million words, most of these are ignored because features are extracted only near possible sentence boundaries. Consider the fragment "... the U.S. Apparently some ...", which the system with only the first four features fails to split after "U.S." The word "Apparently" starts only 8 sentences in the training data, but since it usually appears lowercased (89 times in training), its capitalization here is meaningful. Feature class 6 encodes this idea, indicating the log count[4] of lowercased appearances of the right context: the higher the count, the better the chances that this uppercase appearance starts a new sentence. Similarly, the more times the left context appears without a period in training, the better the chances that this is not an abbreviation.

Another way to incorporate all of the training data is to build a model of $P(s|"L\ R")$, as is often used in sentence segmentation for speech recognition. Without a period in

[4]The log count is rounded to the nearest integer value.

the conditional, many more negative examples are included in the training set. The resulting SVM model is very good at placing periods given input text without them (0.31% error rate), but when limiting the input to examples with ambiguous periods, the error rate is not competitive with our original model (1.45%).

Corpus	**Examples**	**in** S	**SVM Err**	**NB Err**
WSJ	26977	74%	0.25%	0.35%
Brown	53688	91%	0.36%	0.45%
Poe	11249	95%	0.52%	0.44%

Table 3.4: SVM and Naive Bayes classification error rates on different corpora using a model trained from a disjoint Wall Street Journal dataset. The fraction of examples that are sentence boundaries is denoted by $in\ S$.

Classification results using a Support Vector Machine[5] and Naive Bayes[6] are summarized in Table 3.4 for three different datasets.

While the Wall Street Journal corpus (WSJ) is limited to news, the Brown corpus includes 500 documents, distributed across 15 genres, intended as roughly representative of all published English; The Complete Works of Edgar Allen Poe includes an introduction, prose, and poetry. Table 3.5 shows how performance on these corpora scale with the number of training examples. Reasonable performance is possible with limited labeled data, and it appears that additional data would continue to yield improvements.

Corpus	**5**	**50**	**500**	**5000**	**40000**
WSJ	7.26%	3.57%	1.36%	0.52%	0.25%
Brown	5.65%	4.46%	1.65%	0.74%	0.36%
Poe	4.01%	2.68%	2.22%	0.98%	0.52%

Table 3.5: SVM error rates on the test corpora, using models built from different numbers of Wall Street Journal training sentences.

While I have no experiments quantifying the improvements in summary quality that result from this improved sentence segmentation, I observed that only one summary out of the 88 submitted for evaluation in 2009 contained a sentence boundary error. The 2008 submission included 96 summaries, 20 of which had at least one sentence boundary error (I used the *Punkt* system).

[5] A simple linear kernel is used.
[6] With add-k smoothing, $k = 0.25$.

Chapter 4

Selecting Sentences

Okay, so we have a set of input documents. They've been selected by a person to match the query we want to address (in principle this may be the job of Information Retrieval), they've been cleaned up and split into sentences. This chapter is concerned with selecting sentences appropriate for a summary.

4.1 Baselines

We'll begin by establishing a few baselines. The standard baseline for our data (B_1) is to select the first L words from the most recent input document, where L is the length limit. A much stronger baseline (B_2) recognizes that in news documents, first sentences tend to summarize well, and simply selects first sentences starting with the most recent document, up to the length limit. Table 4.1 shows some ROUGE scores for B_1 and B_2. The automatic metrics of choice will be ROUGE-2 and ROUGE-SU4, since their correlation tends to be lower than any other pair of ROUGE metrics.

	ROUGE-2		**ROUGE-SU4**	
Data Set	$\mathbf{B_1}$	$\mathbf{B_2}$	$\mathbf{B_1}$	$\mathbf{B_2}$
2006	0.060	0.070	0.108	0.122
2007	0.059	0.094	0.106	0.143
2008	0.060	0.084	0.091	0.119
2009	0.063	0.092	0.099	0.127

Table 4.1: Baseline ROUGE results for four datasets. B_1 selects sentences in order from the most recent document up the length limit; B_2 selects first sentences only up to the length limit. Note that the 2006 and 2007 are 250-word summaries of 25 documents while 2008 and 2009 are 100-word summaries of 10 documents.

These scores are not particularly meaningful by themselves, but serve as a reference point for further comparison. For now, note how large the gap is between B_1 and B_2: over 20% improvement in most cases[1].

4.2 Maximum marginal relevance

Imagine you are asked to generate a summary by selecting sentences. You read the topic and the narrative description and scan through the documents. Perhaps you choose a sentence that responds well to the query and put it in your summary. How to select a second sentence? You'd like it to respond to the query as well, but don't want it to contain the same information as the first sentence you chose. After all, you are working under a tight length constraint.

Maximum Marginal Relevance [Carbonell and Goldstein, 1998; Goldstein *et al.*, 2000] is an algorithm that formalizes this intuition: maximize relevant information and minimize redundancy. Here's their equation for choosing the next sentence:

$$\text{MMR} = \operatorname*{argmax}_{S_i \in D \setminus S} \left(\lambda(Sim_1(S_i, Q)) - (1-\lambda) \max_{S_j \in S}(Sim_2(S_i, S_j)) \right) \tag{4.1}$$

Initially developed for Information Retrieval, the formulation applies to summarization with R, the set of all input sentences, S, the set of sentences already selected for the summary, Q, a query, and Sim_1 and Sim_2, functions that return the similarity between two sentences. At each step, the MMR sentence is the highest scoring remaining sentence: maximally similar to the query and penalized by its similarity to its nearest neighbor in the selected set. With $\lambda = 1$, the algorithm chooses all relevant sentences, without regard to redundancy; with $\lambda = 0$, relevance is of no importance and selected sentences are as dissimilar as possible—maximizing coverage. To create a summary with MMR, sentences are selected greedily until the length limit is reached. Ideal values of λ appear to vary widely depending on the source documents.

To fully specify MMR, we need to choose similarity functions. The standard similarity function, also borrowed from Information Retrieval, is the cosine distance between the word vectors for each sentence where the value for each word is the Term Frequency Inverse Document Frequency (TFIDF). In my experiments, however, equivalent performance was obtained using a simple word overlap measure: The number of shared non-stopwords normalized by the length of the longer sentence.

Table 4.2 shows scores for summaries generated with MMR with the optimal λ^*

[1]Statistical significance at 95% confidence for these data requires a gap of roughly 0.005 in ROUGE (the standard toolkit uses bootstrapping, sampling sets of 3 out of 4 references, to estimate score variances).

value (shown for MMR_1). The cosine distance function (with TFIDF) is used for both similarity functions. MMR_1 considers all the input sentences, while MMR_2 is limited to the set of first sentences.

		ROUGE-2		**ROUGE-SU4**	
Data Set	λ^*	$\mathbf{MMR_1}$	$\mathbf{MMR_2}$	$\mathbf{MMR_1}$	$\mathbf{MMR_2}$
2006	0.8	0.085	0.079	0.138	0.130
2007	0.8	0.100	0.106	0.150	0.155
2008	1.0	0.076	0.100	0.113	0.128
2009	0.9	0.084	0.097	0.119	0.132

Table 4.2: Maximum Marginal Relevance scores: MMR_1 selects from all input sentences; MMR_2 selects from the set of first sentences.

While MMR gives reasonable improvements over the baselines, the first-sentence version remains stronger overall. This suggests that MMR, or perhaps our choice of similarity function, is not powerful enough to make productive use of the full set of input sentences.

4.3 Using document statistics

It turns out that MMR is broken in a number of ways. First of all, the query in these problems is too limited to allow for meaningful measurement of similarity. In particular, many valuable sentences do not overlap substantially with the the query (in word usage). One option is to follow the course set by Information Retrieval and try expanding the query [Goldstein *et al.*, 1999]. Still, relying on the query as the seed for importance is usually too risky. Better results can be obtained by ignoring the query altogether.

This may be a product of these datasets: because the input documents are hand-picked to match the query, a more reliable signal can be extracted from the document set than from the query alone. Still, it seems likely that the underlying structure in any query-derived document set would convey a stronger signal than the query alone; a systematic study of this claim would be useful.

In general, the shift away from Information Retrieval approaches has been productive for summarization. This trend was demonstrated nicely by Nenkova and Vanderwende when they introduced SumBasic in 2005, a simple system based on word frequency in the document set that outperformed most everything more complex at the time [Nenkova and Vanderwende, 2005; Nenkova *et al.*, 2006].

Figure 4.1 shows how frequency in the input documents effects the probability that a word makes it into at least one of the references. The SumBasic algorithm tries to build a summary that includes lots of high-frequency words. It runs as follows:

1. Estimate the unigram distribution from the input documents: $p(w_i) = \frac{\text{count}(w_i)}{N}$, where N is the total number of words in the documents.

2. For each sentence S_j assign a weight equal to the average probability of the words it contains: $weight(S_j) = \sum_{w_i \in S_j} \frac{p(w_i)}{|w_i \in S_j|}$

3. Pick the highest scoring sentence that also contains the best scoring word and add it to the summary (so long as it fits within the length limit).

4. For each word in the sentence selected in step 3, update its probability: $p_{\text{new}}(w_i) = p_{\text{old}}(w_i) * p_{\text{old}}(w_i)$

5. If the desired summary length has not been reached, go to step 2.

Bigrams work better than unigrams (if both words are stopwords, the bigram is ignored), document frequency works better than raw frequency, and making sure the most valuable bigram is in the selected sentence actually degrades performance. Table 4.3 shows modified SumBasic (SumBasic+) results on our data. As above, we show two versions: SumBasic+$_1$ uses the full set of input sentences while SumBasic+$_2$ uses only first sentences. These results are significantly better than the MMR results, and at last, we are able to match or exceed the first-sentences-only version using the full set of sentences.

	ROUGE-2		**ROUGE-SU4**	
Data Set	**SumBasic+$_1$**	**SumBasic+$_2$**	**SumBasic+$_1$**	**SumBasic+$_2$**
2006	0.092	0.083	0.140	0.132
2007	0.119	0.113	0.165	0.158
2008	0.097	0.104	0.130	0.133
2009	0.103	0.101	0.134	0.133

Table 4.3: The modified version of SumBasic

SumBasic improves on MMR by estimating relevance directly from the input documents, rather than mediated through the query. But it also improves on MMR in its treatment of redundancy. Rather than model redundancy explicitly, by trying to select sentences that overlap minimally with those already selected, there is an implicit redundancy constraint: words lose value once they've been added to the summary. Without the probability updates (skipping step 4), average ROUGE scores decrease by 6-18%.

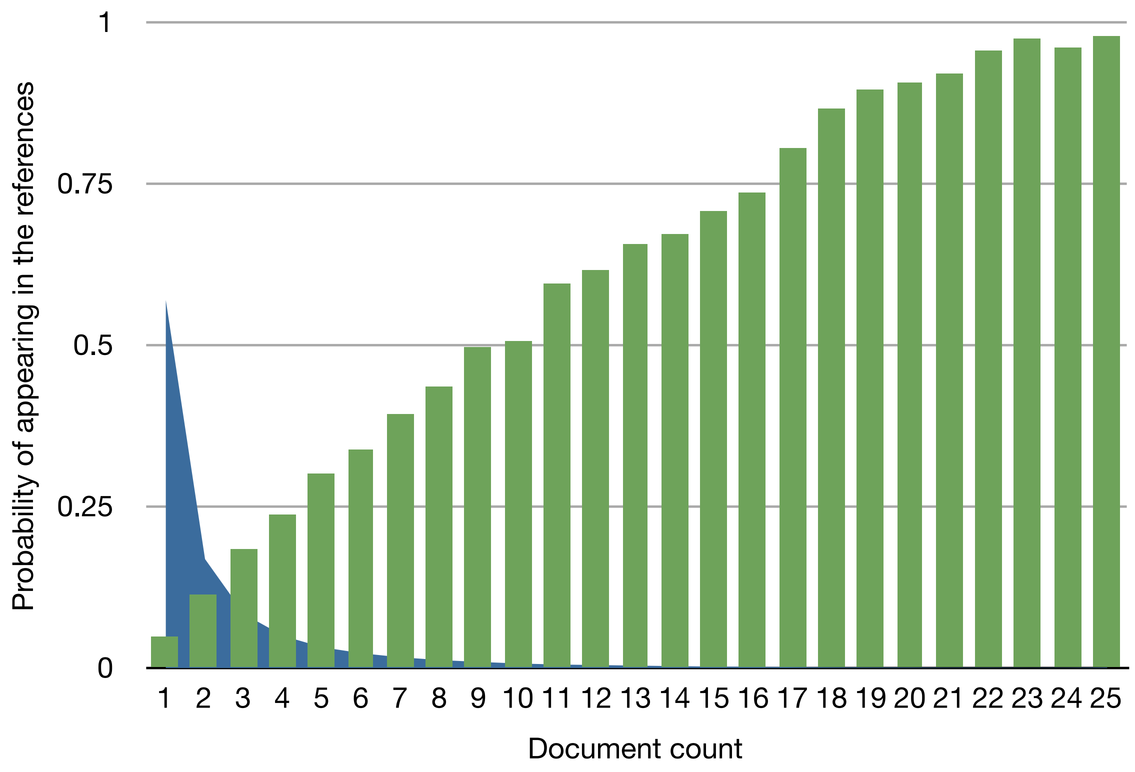


Figure 4.1: The number of documents in which a word appears (stopwords excluded) has a strong (nearly linear) relationship with the estimated probability that word appears in at least one of the reference summaries. The blue area shows the fraction of words with each count: only around 10 words per document set appear in 20 or more of the 25 documents. 2006 and 2007 data.

Remarkably, SumBasic+ produces summaries that are nearly state-of-the-art (see figure 4.2). Adding a heuristic to upweight (by a factor of 2) n-grams appearing in the first sentence of a document yields a simple system (SumBasic++) that outperforms all other (typically far more complex) systems submitted to the annual evaluations, at least as measured by ROUGE.

4.4 From procedure to objective

At each iteration, SumBasic chooses the most valuable remaining sentence—a greedy search of some kind. But what are we searching for? The goal is never defined, so SumBasic is just a procedure. It works well but it doesn't allow us to identify a model of what makes a good summary.

Instead, let's frame the process as an objective function on summaries. First, we observe that SumBasic's down-weighting may as well be zeroing (this switch indeed has a minimal effect on the resulting scores). Then, the value S of an entire summary is simply the sum of the n-gram values it contains, except that credit is only given to each n-gram once, an implicit constraint on redundancy replacing MMR's clunky pairwise similarity estimate. This idea specifies a model for summarization. Here is some notation:

$$S = \sum_i w_i c_i$$

To remain as general as possible, let's replace "n-grams" with "concepts": Concepts are indexed by i, with c_i an indicator for the presence of concept i in the summary; w_i is the weight, or value, of concept i.

Now, assuming we are given a set of weights, we can search for the summary that maximizes the objective function, subject of course to a constraint on word length L. Since we are limiting the summary to extracted sentences, let's describe the length constraint in terms of selected sentences. As with the concepts, sentences are indexed by j, with s_j an indicator for the presence of sentence j in the summary; l_j is the length (in words) of sentence j.

$$\begin{aligned} \text{Maximize: } & \sum_i w_i c_i \\ \text{Subject to: } & \sum_j l_j s_j \leq L \end{aligned}$$

One way to understand this objective is as a weighted maximum coverage problem. The input space contains weighted concepts, blocked into overlapping units of selection (sentences). We want to find the set of sentences that covers as much of the space as possible subject to a constraint on the sum of the sentence lengths. This maximum coverage formulation was introduced for summarization some time ago [Filatova and Hatzivassiloglou, 2004], but the focus was on entity extraction rather than useful objective functions, and they used a greedy approximation.

Searching for the best solution to this problem will most likely require an algorithm exponential in the number of input sentences: A reduction from the well-known set-cover problem [Hochbaum, 1996] is straightforward. Or, consider the NP-complete knapsack problem: maximize the value of items in a knapsack, where each item has a value and a weight, and the knapsack has a weight limit. In the case where our sentences have no overlapping concepts, it is precisely the knapsack problem (each

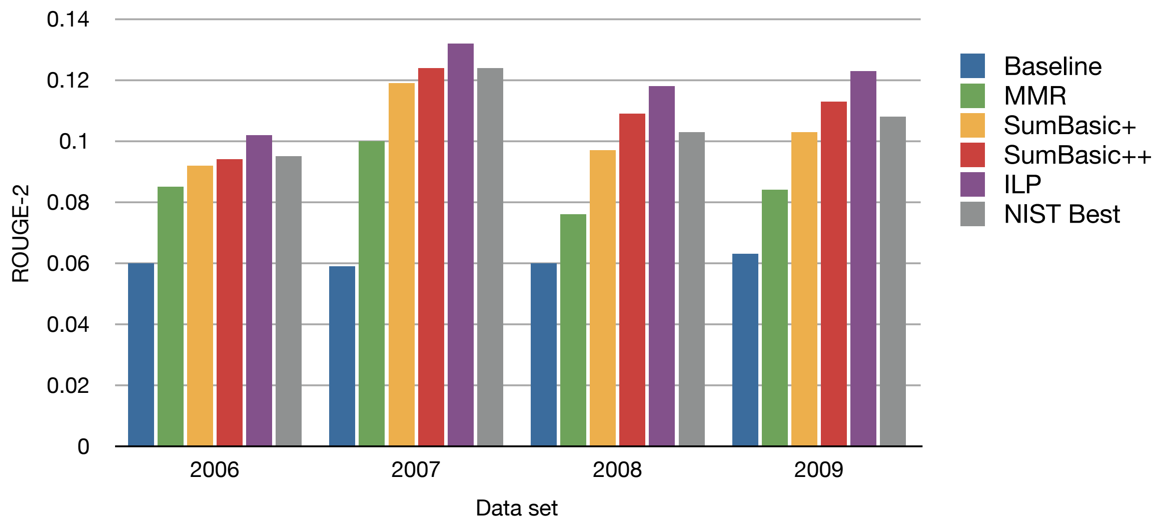


Figure 4.2: ROUGE-2 scores for the systems discussed in this section. The best NIST scores are shown for comparison (excluding systems submitted by me). Recall that SumBasic+ refers to my enhanced version of the original SumBasic algorithm, and SumBasic++ adds a heuristic to increase the weight of n-grams appearing in first sentences.

sentence is an item and the summary is the knapsack). Thus the more general case, where concepts are shared among sentences, reduces to the knapsack problem in this limited case. As a result, a polynomial time solver for our problem could surely solve the knapsack problem in polynomial time, so our problem is also NP-complete.

The next chapter will address approximate and exact inference in this framework. The results shown in Figure 4.2 include an exact solution to the maximum coverage objective, obtained using an Integer Linear Program (ILP) solver. The maximum coverage system is otherwise identical in its setup to SumBasic++ (the value w_i of bigram i is it's document count, with upweights for bigrams appearing in first sentences), except that concepts appearing in fewer than three documents are ignored for efficiency.

Since I submitted the maximum coverage system, with only a few small enhancements [Gillick *et al.*, 2010], to the 2009 evaluation, these summaries were evaluated by experts. Figure 4.3 compares its performance to all the other submitted systems. Such strong results are good evidence that our model is reasonable: Even though the objective is extremely simple, the scores are substantially better than other, far more complex systems.

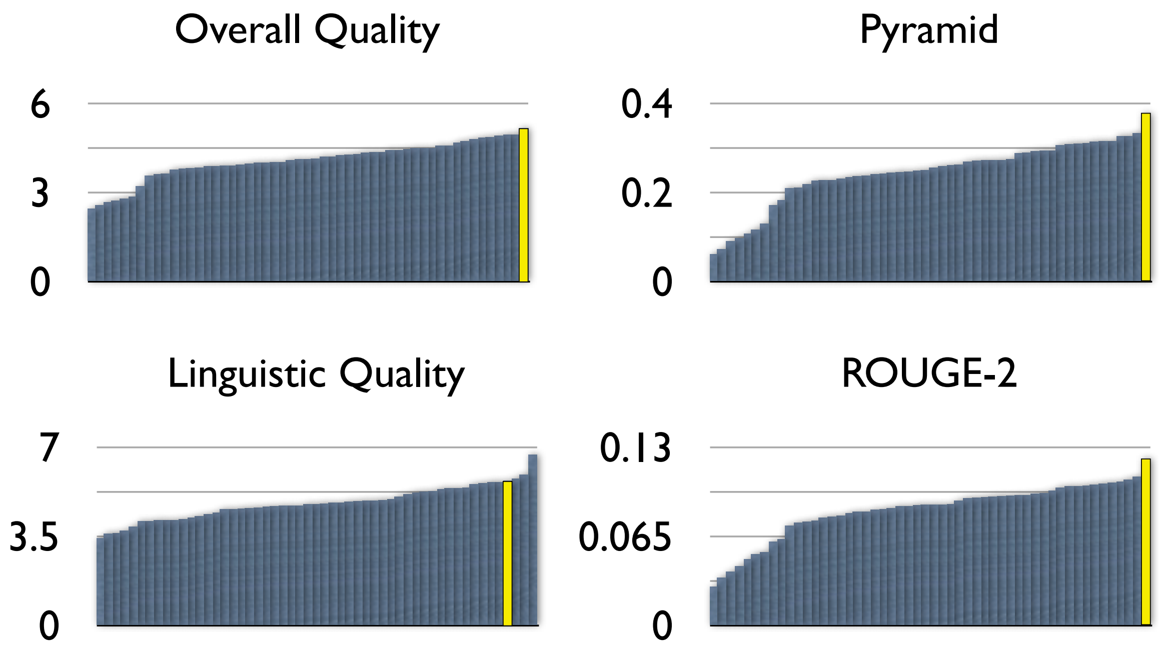


Figure 4.3: 2009 results for a variety of metrics: The maximum coverage system (with a few small enhancements) is the yellow bar; the blue bars show scores for all the other systems submitted to the evaluation. Note that the standard baseline (first 100 words of the most recent document) is included, and has the highest Linguistic Quality score.

4.5 More ideas in extraction

So far, I've created a very linear narrative for progress in summarization research. A number of other ideas in sentence selection co-evolved and merit some discussion.

4.5.1 Learning sentence values

What makes for a good summary sentence? Lots of frequently occurring words, position in the document, similarity to a query? If we had training data—input sentences labeled as summary or non-summary—we could run standard machine learning algorithms to learn weights for these features.

Since the data includes reference abstracts rather than extracts, we need to choose sentence labels before learning. While some studies employ a classification framework with binary sentence labels, regression seems a more natural choice, so we'll set the label of each sentence to be the average number of reference bigrams it contains. We treat the four references given for each problem as samples from the space of all possible references, so if a bigram appears in three of them, the maximum likelihood

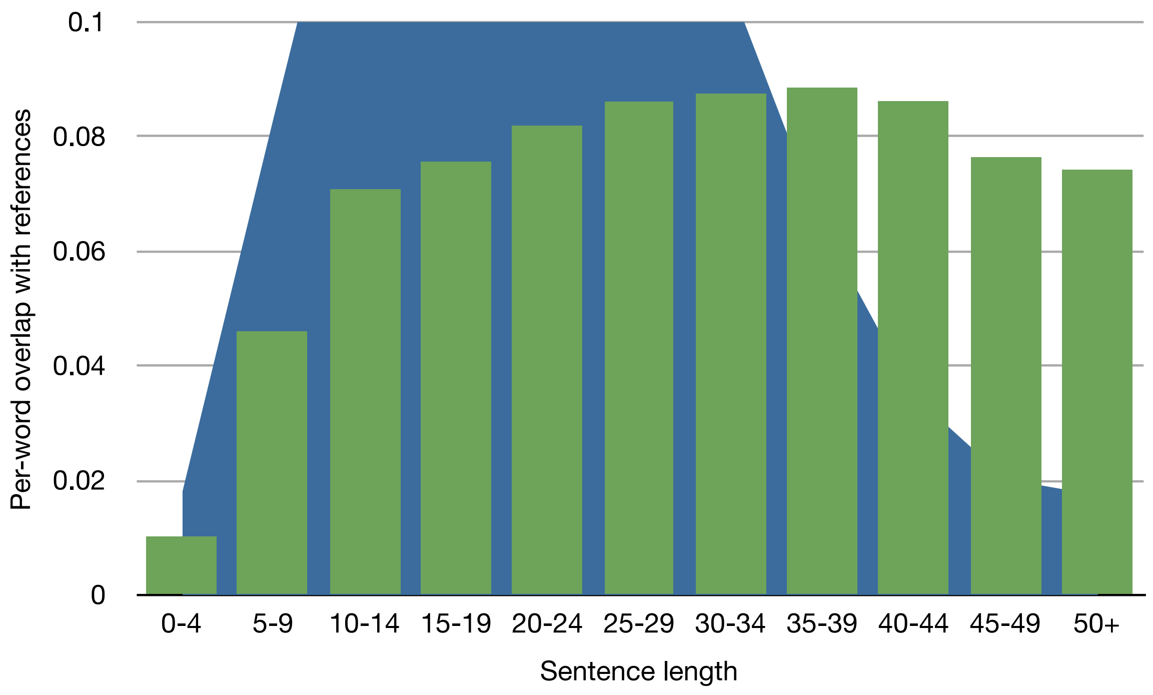


Figure 4.4: The value of a sentence increases sharply as its length (in words) increases, but then begins to fall off if the sentence gets too long (over 40 words). The blue area shows the fraction of sentences in each length bin. 2006 and 2007 data.

estimate of the probability it appears in a random reference is $\frac{3}{4}$.

I experimented with a variety of regression models, including SVM regression as used by Schilder and Kondadadi [2008], linear regression, and Poisson regression. Poisson regression assumes the response variable has a Poisson distribution and thus is especially well-suited to model counts. While Poisson regression appears to fit the training data best, linear regression gave comparable or slightly better ROUGE scores when a summary is built using the top-scoring sentences[2].

The following set of simple features were used as predictors in the regression, and an analysis of predictive power is shown in Table 4.4. Figures 4.4, 4.5, 4.6 give a more fine-grained analysis for some features.

Frequency average per-word document frequency

Topic average per-word topic description overlap

[2]Linear regression has the undesirable property of producing a few negative expected counts, which we force to zeros.

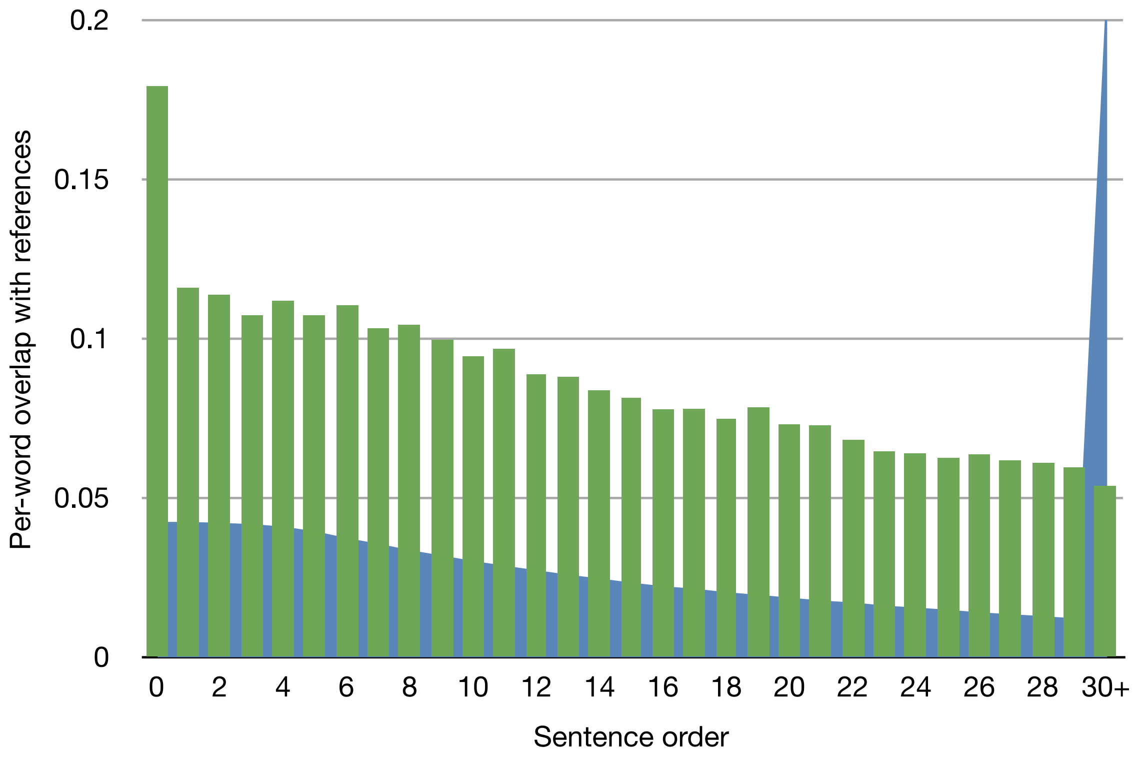


Figure 4.5: Sentence value and position in the document are strongly related. Value declines roughly monotonically (some noise here), but the first sentence is in a class of its own. The blue area shows the fraction of sentences of each length, which declines because not all documents are of equal length. 2006 and 2007 data.

First Sentence indicator for the first sentence in a document

Sentence Position log of the sentence position in the document

Sentence Length log of the sentence length in words

Document Length log of the document length in sentences

In the table, the constant term estimates the mean, 0.75 reference words per sentence, and average document frequency is the strongest individual feature. Document length, which I've never seen used as a feature before, is interesting: longer documents have fewer reference words per sentence. Using all features together gives a Root Mean Squared Error (RMSE) of 0.85, a 36% improvement over the constant term baseline. Modeling pairwise products between features improves the RMSE to 0.79, a total improvement of 47%.

While the predicted sentence values given by regression are reasonable, using the predictions to create a summary is not straightforward. Most documented work along

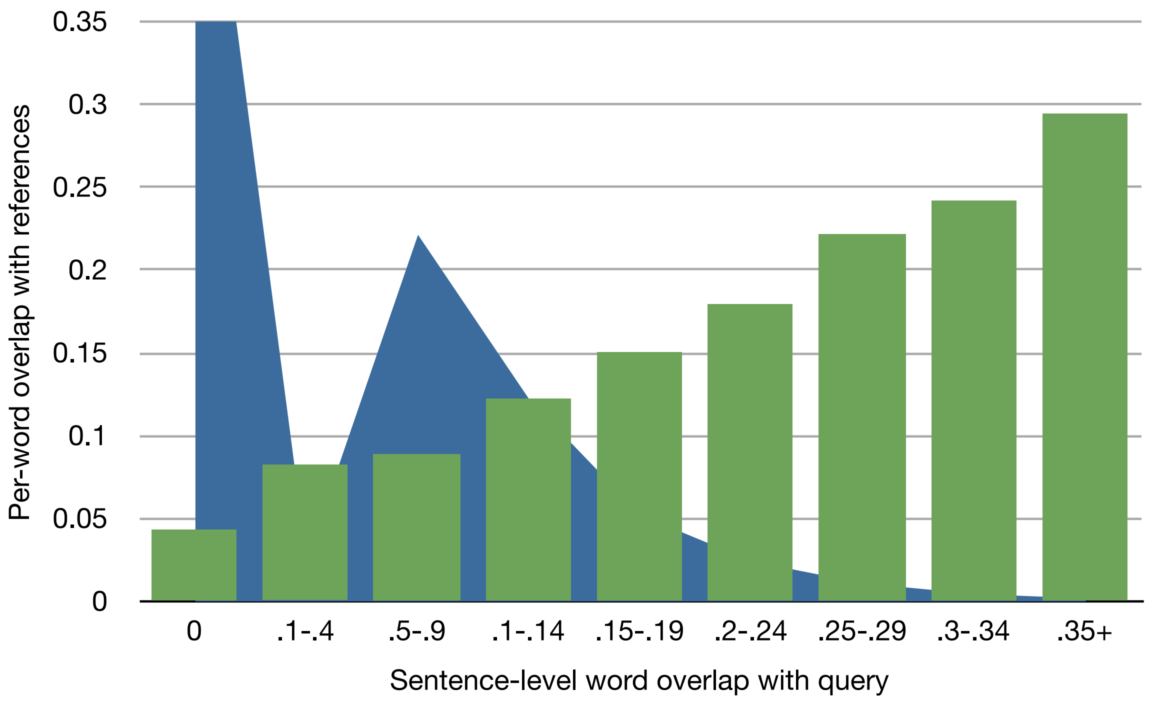


Figure 4.6: The more a sentence overlaps (fraction of content words) with the query, the more valuable it tends to be. The blue area shows the fraction of sentences in each bin: most sentences have no overlap but the few sentences with very high overlap have considerably higher value than even first sentences. 2006 and 2007 data.

these lines involves using MMR-like criteria to maximize relevance and minimize redundancy. Treating the learned sentence values as priors to help weight bigram counts, the inputs to the maximum coverage system, can give some small improvements.

But this sort of training is fundamentally flawed because the learning happens at an intermediate stage of summarization: It is not obvious how to create high ROUGE summaries from high ROUGE sentences. Still, nearly all supervised learning for summarization has been along these lines, with various takes on regression, classification, and algorithms for minimizing redundancy during sentence selection [Kupiec *et al.*, 1995; Conroy and O'Leary, 2001; Shen *et al.*, 2007; Li *et al.*, 2007].

Note that Yih et al. [2007] learn word probabilities with logistic regression (as opposed to sentence values). While this leads to a more natural criteria for creating summaries with maximum total word value (similar to my ILP formulation), the learning still suffers from the same mismatch problem, ignoring the underlying relationships between sentences and words.

Feature	Estimate	CV RMSE	% gain
Constant	0.75	1.16	
Frequency	0.49	0.96	20.6%
Topic	0.18	1.07	8.5%
First Sentence	0.15	1.10	5.2%
Sentence Position	-0.09	1.09	6.3%
Sentence Length	0.28	1.09	6.3%
Document Length	-0.04	1.13	2.7%
All		0.85	36.4%

Table 4.4: Analysis of a linear regression predicting the expected number of reference bigrams in a sentence. Each feature is normalized to have mean 0 and variance 1. CV RMSE is the cross-validation root mean squared error of the predictions, with 90% train 10% test splits. 2008 data.

Learning feature weights that directly maximize ROUGE or some other automatic metric given, for example, the maximum coverage objective, is a task for structured prediction: The learning algorithm tries to select a set of weights that actually recover the ideal training summaries. Some of my more recent work [Berg-Kirkpatrick *et al.*, 2011] addresses this problem.

4.5.2 Graph algorithms

So far, we've treated the set of input documents as a bag of words (bigrams, really). But perhaps we'd benefit from respecting some of the structure that connects words and sentences, for example. We can construct a graph with a vertex for each word and each sentence with edges connecting a sentence to the words it contains. A variety of similar graph structures are employed in the literature [Erkan and Radev, 2004; Mihalcea and Tarau, 2004]. Then we can use an iterative Pagerank-style algorithm [Page *et al.*, 1998] to estimate the value or relevance of each word and each sentence as follows (see Figure 4.7):

Initialization: Maintain a distribution over word values and a distribution over sentence values, initialized uniformly.

Iteration Step: The new value of a word is the sum of the sentence values it's connected to; the new value of a sentence is likewise the sum of word values it's connected to. The distributions are normalized.

This algorithm is typically allowed to run until convergence, though sometimes with a damping factor that prefers uniformity. Intuitively, sentences with the highest

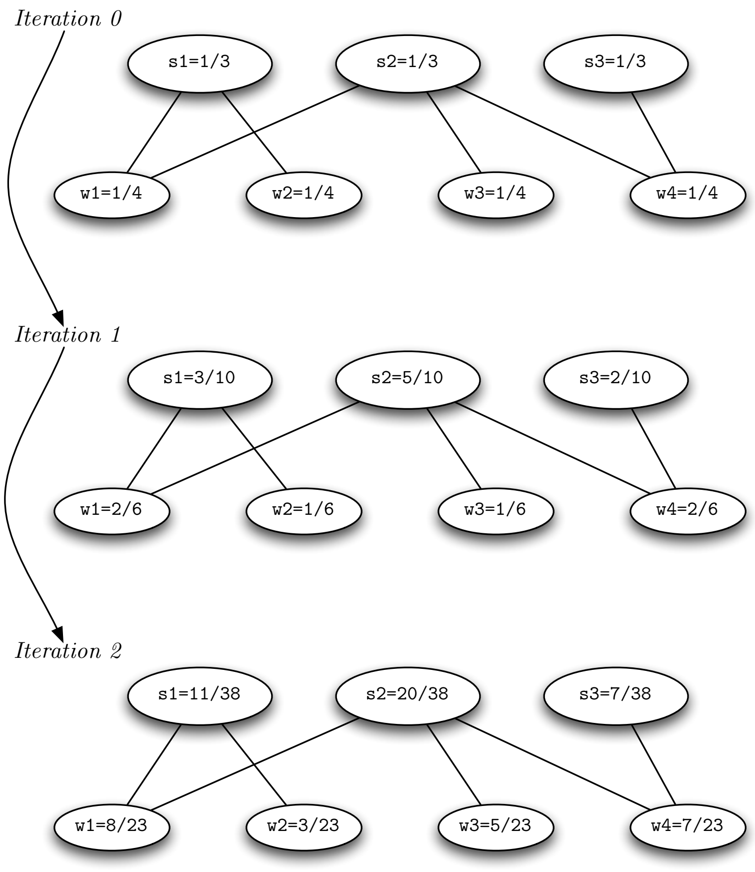


Figure 4.7: A graphical depiction of a simple graph algorithm over three sentences and four words. At iteration 0, the values are initialized to be uniform, and by iteration 2, the values have nearly converged.

values are most central to the collection. In fact, there is a theoretical interpretation for the resulting values: If we take a sufficiently lengthy walk through the graph, moving at random along edges, the normalized values represent the probability of stopping at some word or sentence vertex.

Of course, while the random walk model may be a reasonable proxy for Internet browsing, it seems less relevant for reading a bunch of documents: Reading (or writing) is mostly linear and doesn't involve jumping between similar sentences. As a result, the decision to use the word values after one iteration (these are just simple counts as in the maximum coverage formulation), is no harder to justify than allowing the algorithm to run to convergence.

Thus, the advantages conferred by a graph representation of a document set may not be realized through the Pagerank algorithm. And in fact, I have not been able to produce any performance gains using any sort of simple iterative procedure like the algorithm described here.

4.5.3 Topic models

All this talk of word values is a little fishy. The value of a word comes more from the context in which it's used than from its inherent meaning or the number of times it appears. And while sentence value makes more sense, sentences come in all different shapes and sizes, with varying amounts of embellishment. Ideally, summarization should concern itself with some more appropriate semantic units: facts and topics seem preferable to words and sentences.

One step in this direction is topic modeling, where a latent set of topics are inferred from word occurrences in a set of documents. The simplest form is Latent Semantic Indexing (LSI): A term-sentence matrix is constructed with a row for each sentence and a column for each content word in the document set; the values in the cells could be binary, counts, or TFIDF values, for example. Running Principal Components Analysis (PCA) on this matrix yields a series of eigenvectors, linear combinations of the words, that capture the most remaining variance. Each resulting component can be considered a kind of topic, and each sentence, projected onto the first few components, has an interpretation as a weighted mixture of the important topics. Gong and Liu [2001] describe summary generation by selecting the sentences that best represent the first few topics.

A simple generative model for topics yields a probabilistic interpretation of LSI [Hofmann, 1999] called probabilistic Latent Semantic Analysis (pLSA). The pLSA generative story explains how a set of documents is made:

1. Pick a document d from D with probability $P(d)$
2. Pick a latent topic z from Z with probability $P(z|d)$
3. Generate a word w of W with probability $P(w|z)$

The parameters of the model can be estimated via the Expectation Maximization (EM) algorithm, which reaches a local maximum of the likelihood function:

$$\mathcal{L} = \sum_{d \in D} \sum_{w \in W} n(d, w) \log \sum_{z \in Z} P(w|z) p(z|d) P(d)$$

where $n(d, w)$ is the number of times word w appears in document d.

By replacing documents with sentences, the topic interpretation matches LSI at the sentence level. Each sentence is represented by a vector in the "topic-space" of the document set. The probability distribution $P(z|s)$ tells us to what extent each topic is covered by a sentence:

$$P(z|s) = (P(z_1|s), P(z_2|s), \ldots, P(z_k|s))$$

A comparative study suggests pLSA may be preferable to LSI for the purposes of summarization [Hennig and Labor, 2009]; at least it has a more intuitive interpretation as a model.

Augmenting the standard pLSA model with a Dirichlet prior on topics gives an improved topic model called Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003; Steyvers and Griffiths, 2007]. The main change to the generative story is that the first word is picked from the first topic and subsequent words are picked from the set of existing topics with probabilities proportional to the sizes of the topics. Since this model generates an entire dataset at once, whereas pLSA generates each word independently, alternate inference procedures are needed. Variational methods or Gibbs sampling are standard[3].

Recent implementations of topic models for summarization have shown good results, [Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010] though I have no head-to-head comparison with the maximum coverage model (ROUGE results suggest comparable performance). Note that Haghighi and Vanderwende use topic models for estimating word values and then proceed to select sentences greedily with a KL-divergence criteria[4]; Celikyilmaz and Hakkani-Tur use topic models to estimate sentence values and are then stuck with the familiar problem of choosing maximally relevant sentences with minimal redundancy.

[3]See Kevin Knight's tutorial on Bayesian inference for a readable introduction: http://www.isi.edu/natural-language/people/bayes-with-tears.pdf

[4]Choose the summary that minimizes the KL-divergence in word distribution between summary and source documents.

Chapter 5

The Maximum Coverage Objective

The last chapter outlined a variety of approaches to valuing sentences and choosing summaries, all reasonably well motivated. Still, as best I can tell, applying the maximum coverage objective over word bigrams gives results at least as good. It is simple and appealing: a summary attempts to cover as many important concepts as possible. Rather than attempt to approximate the redundancy between two sentences with a function, redundancy is treated implicitly: the summary only receives credit once for each concept.

This chapter describes methods for solving the maximum coverage objective, approximately and exactly.

5.1 Approximate and exact solving

Recall that maximum coverage seeks to maximize $\sum_i w_i c_i$, where c_i is an indicator for the presence of concept (bigram) c in the summary. The value w_i for bigram i is estimated simply as the document count of c_i, with an extra count for each first sentences it appears in. Of course, these values could be estimated in some more principled manner, but for present purposes, I'll focus just on solving the objective regardless of the source of the values.

The simplest way to generate a summary from the objective is to choose one sentence at a time that maximizes the objective value so far. There are really two ways to do this in our case. The value added by a sentence could be (1) the actual sum of bigram values in the sentence that do not appear in the partial summary, or (2) the value of (1) normalized by the by the length of the sentence. In practice, method (2) usually gives higher total objective values since method (1) tends to over-prefer long sentences. Our greedy algorithm generates a summary with both (1) and (2) and chooses the one with the larger objective value.

Exact solutions can be obtained, as suggested earlier, with a formulation of the objective as an Integer Linear Program [Nemhauser and Wolsey, 1988]. While ILP solutions are exponential in the size of the input, solvers have been highly optimized to give fast solutions for many problems. Here is the full ILP specification of our objective:

$$\text{Maximize: } \sum_i w_i c_i \tag{5.1}$$

$$\text{Subject to: } \sum_j s_j l_j \leq L \tag{5.2}$$

$$s_j Occ_{ij} \leq c_i, \ \forall i, j \tag{5.3}$$

$$\sum_j s_j Occ_{ij} \geq c_i, \ \forall i \tag{5.4}$$

$$c_i, c_j \in \{0, 1\}, \ \forall i, j \tag{5.5}$$

(5.1) and (5.2) we have seen before; (5.3) and (5.4) are structural constraints ensuring that sentences and their concepts are consistent. Occ is a matrix with Occ_{ij} an indicator for the presence of concept i in sentence j. Thus, (5.3) requires that selecting a sentence entails selecting all the concepts it contains, and (5.4) requires that selecting a concept is only possible if it is present in at least one selected sentence.

Data Set	**Greedy**	**Exact (ILP)**	**Gain**
2008	0.111	0.117	5.4%
2009	0.115	0.121	6.0%

Table 5.1: Comparing average ROUGE-2 scores of greedy and exact solutions to the maximum coverage objective.

Table 5.1 compares ROUGE scores of greedy and exact solutions. While standard ILP solvers give solutions quickly for any of the 10- or 25-document summarization problems addressed here, exact solutions may not always be feasible. When the input space is prohibitively large, approximate solvers can help bridge the gap between greedy and exact solutions. Experiments with a local search algorithm that maintains a beam of hypotheses, swapping random sentences and updating the beam on each iteration, were able to give ROUGE scores on par with the ILP. This is somewhat similar in spirit to the stack-based search employed by Yih et al. [Yih *et al.*, 2007].

5.2 Integer linear programs

I have shown that the mysterious ILP solves this the maximum coverage objective quite handily (the section on ILP run-time, below, quantifies its miraculous performance). To get a sense for how hard the problem is that I'm asking the ILP to solve, consider a typical problem from the 2007 dataset: Choose the set of about 10 sentences (250 words) from the roughly 500 sentences in the document set that gives the best objective score. This gives a lower bound of about $\binom{500}{10} = 2.5 \times 10^{20}$ possible summaries. This is a lower bound because the length limit is over words, so there are many combinations of 8, 9, 10, 11, 12 (etc.) sentences that have under 250 words. Still, this is a lot of summaries to consider. A brute-force enumeration would take hundreds if not thousands of years on a single fast computer.

Certainly, the problem has some internal structure that can make it more tractable. Some sentences may have no relevant concepts; some may be full subsets of others in the space of concepts. Such observations could allow us to create a more efficient algorithm for the specific problem of optimizing this particular objective function on sentences containing concepts. But remarkably, the ILP has no such knowledge. It can be applied to a wide variety of optimization problems, often with impressive results.

A linear program is simply one that can be expressed in this canonical form, with c the weight vector applied to the non-negative vector to be determined x, subject to linear equality or inequality constraints expressed in matrix form (the elements of b are also non-negative):

$$\begin{aligned} \text{Maximize: } & \mathbf{c}^T\mathbf{x} \\ \text{Subject to: } & A\mathbf{x} <= \mathbf{b} \end{aligned}$$

Linear programs were developed in the late 1930s and early 1940s and initially used during World War II to optimize army expenditures. As a result, it was kept secret until 1947, and was immediately adopted by industry to aid financial decisions.

George Danzig's original approach to solving linear programs is still common today. The *simplex* algorithm finds a feasible point at a vertex of the convex shape defined by the constraints and walks along edges to other vertices with non-decreasing objective function values until a maximum value is reached. While this method worked well in practice (and on random problems), it was not until the late 1970s that a different class of algorithms showed that solving linear programs is polynomial even in the worst case. The complexity is roughly $O(n^{3.5})$.

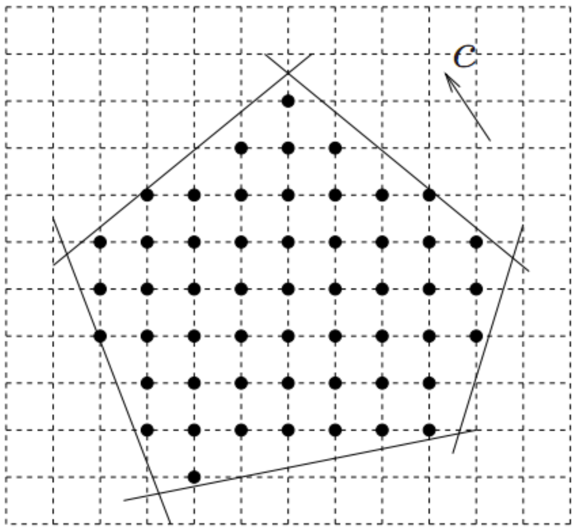


Figure 5.1: A graphical representation of the general ILP problem.

If, however, the unknown variables $\mathbf{x}$ are required to be integers, things get a bit trickier: complexity is NP hard in the worst case. But since the 1950s, techniques for solving these integer-only versions of linear programs, ILPs, have led to increasingly fast solutions for real problems. The oldest such algorithm is Ralph Gomroy's *cutting plane* method, which works by first solving the non-integer relaxation of a problem, and then iteratively adding constraints induced by the separation or *cut* between this optimum and the feasible set. At each iteration, the non-integer solution has more and more integral values. Modern systems often use a hybrid of *cutting plane* and the *branch and bound* method, a way of developing alternative sets of cutting planes to avoid potential numerical instability. This is essentially the default method employed by the ILP solver (GLPK) I used for optimizing the maximum coverage objective.

Many natural language processing tasks lend themselves naturally to ILP formulations. Casting a problem as an ILP allows us to largely ignore the specific hard optimization required for exact solutions. As a result, ILPs have been used to find globally optimal solutions for producing phrase tables for machine translation, finding sentence compressions, inferring document structure, and many others.

5.3 Theoretical guarantees

While the ILP solutions are appealing, our greedy approximation to the maximum coverage objective has an interesting theoretical guarantee: the greedy solution must be within a constant factor $(1 - 1/e)$ of the optimal solution (e is the natural logarithm). While we have shown that the gap between greedy and exact solutions is in fact substantial in our case (see Table 5.1), the nature of this particular theoretically property, *submodularity*, has received considerable attention, and deserves some elaboration.

Let f be a function on sets that maps subsets $S \subseteq V$ to real numbers. $f(\cdot)$ is called *monotone* if $f(S) \leq f(T)$ whenever $S \subseteq T$, and *submodular* if for any $S, T \subseteq V$:

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$$

An equivalent definition of submodularity is the property of *diminishing returns*. That is, $f(\cdot)$ is submodular if for any $R \subseteq S \subseteq V$ and $s \in V \setminus S$,

$$f(S \cup \{s\}) - f(S) \leq f(R \cup \{s\}) - f(R)$$

This says that the value added by s in the larger set S is no larger than the value added by s in the smaller subset R.

Submodularity is a discrete analog of convexity [Lovász, 1983]; both properties tend to facilitate optimization. While maximization of submodular functions is NP-complete, Nemhauser et al. showed famously that monotone submodular functions with a cardinality constraint can be solved with a greedy algorithm within a constant factor $(1-1/e)$, around 63%, of the optimal solution [Nemhauser *et al.*, 1978]. Krause and Guestrin [2005] give an improved bound $\frac{1}{2}(1-1/e)$ for budgeted maximum coverage problems, like ours. Note that including the two variants in our greedy algorithm is necessary for receiving the benefits of submodularity.

The takeaway message from this discussion is that should we endeavor to summarize giant documents or sets of documents where an exact solution is infeasible, our greedy approximation for optimizing the maximum coverage objective is guaranteed to give a solution within about 30% of the optimal solution. Lin and Bilmes [2010] have a more extensive analysis of submodular functions for summarization.

5.4 ILP run-time

While ILPs have been used extensively for natural language processing problems, designing an objective to allow fast solutions can be tricky. To assess performance of our ILP, let's compare the solving time to an adaptation of maximum marginal relevance to an ILP. In particular, McDonald [2007] formulates MMR as follows:

$$\text{Maximize: } \sum_j s_j Rel(j) - \sum_{j' \leq j} \alpha_{jj'} Red(j, j') \tag{5.6}$$

$$\text{Subject to: } \sum_j s_j l_j \leq L \tag{5.7}$$

$$\alpha_{jj'} - s_j \leq 0, \ \forall j, j' \tag{5.8}$$

$$\alpha_{jj'} - s_{j'} \leq 0, \ \forall j, j' \tag{5.9}$$

$$s_j + s_{j'} - \alpha_{jj'} \leq 0, \ \forall j, j' \tag{5.10}$$

$$s_j, \alpha_{jj'} \in \{0, 1\}, \ \forall j, j' \tag{5.11}$$

Given some definitions of relevance (Rel) and pairwise redundancy (Rel), this version of MMR gives a globally optimal selection of sentences subject to a length constraint. Because redundancy is defined over sentence pairs, McDonald coerces a kind of quadratic knapsack problem [Gallo *et al.*, 1980] into an a linear program by introducing indicators $\alpha_{jj'}$ for selecting a pair of sentences j and j'. As in the original paper, relevance and redundancy are computed as follows:

$$Rel_j = cosine(j, D) + 1/pos(j, D)$$
$$Red_{jj'} = cosine(j, j')$$

Here, $cosine(j, D)$ is the cosine distance between the TFIDF vectors for sentence j and document D, and $pos(j, D)$ is the integer-valued order of sentence j in document D.

With n input sentences and m concepts (bigrams), both formulations generate a quadratic number of constraints. However, the MMR formulation has $O(n^2)$ variables while the maximum coverage formulation has $O(n + m)$. In practice, scalability is largely determined by the sparsity of the redundancy matrix Red and the sentence-concept matrix Occ. Efficient solutions thus depend heavily on the redundancy measure and the concept unit. Pruning to reduce complexity involves removing low-relevance sentences or ignoring low-redundancy values in the MMR formulation, and corresponds to removing low-weight concepts in the maximum coverage formulation. Note that pruning concepts may be more desirable: Pruned sentences are irretrievable, but pruned concepts may well appear in the selected sentences through co-occurrence with unpruned concepts.

Figure 5.2 compares ILP run-times for the two formulations, using a set of 25 problems from the 2007 dataset, each of which have at least 500 input sentences. While the ILP solver[1] finds optimal solutions efficiently for the maximum coverage

[1]I use GLPK with all the default settings (gnu.org/software/glpk)

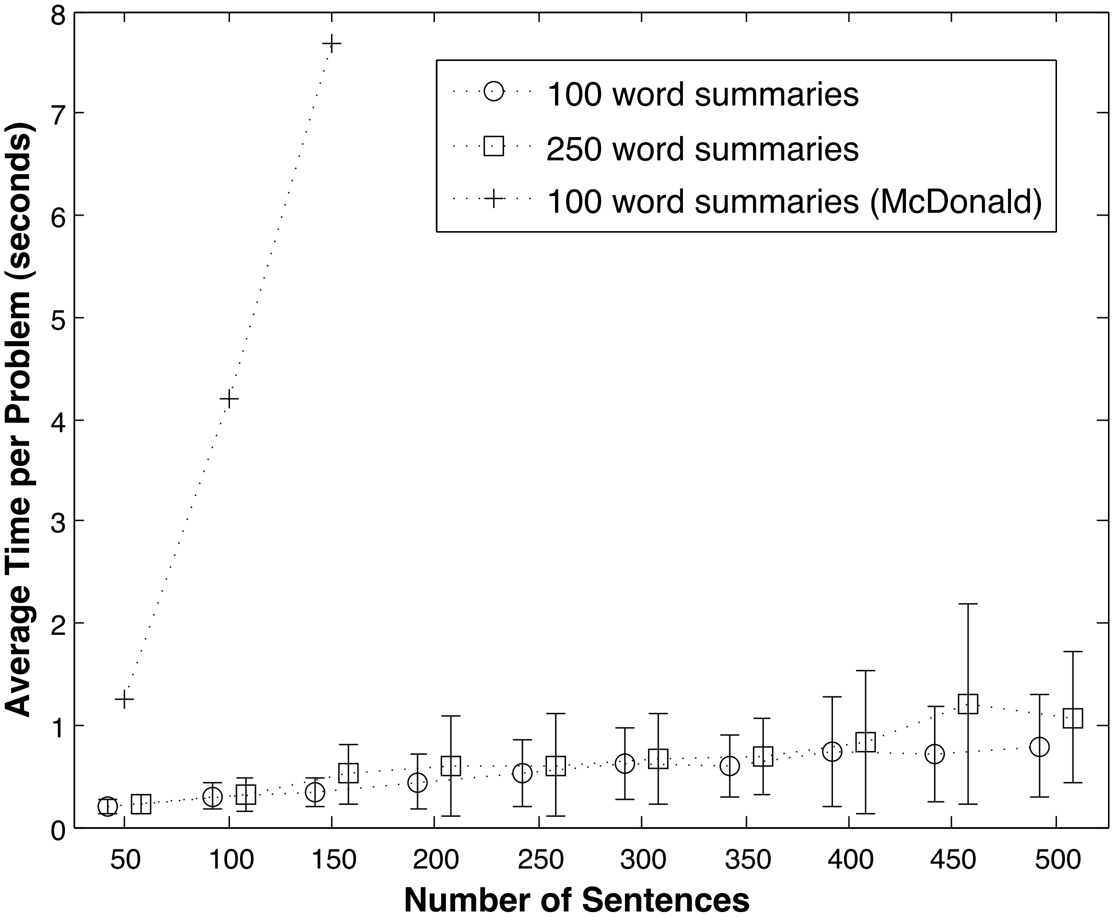


Figure 5.2: A comparison of ILP run-times (on an AMD 1.8Ghz machine) of McDonald's MMR formulation and the maximum coverage formulation on an increasing number of input sentences.

formulation, the MMR solution times grow rapidly with the size of the input. The plot includes timing results for both 100 and 250 word summaries, showing that fast solutions are given even for much harder problems: A rough estimate for the number of possible summaries has $\binom{500}{4} = 2.6 \times 10^9$ for 100-word summaries and $\binom{500}{10} = 2.5 \times 10^{20}$ for 250 word summaries.

The maximum coverage objective also gives far better ROUGE scores than McDonald's MMR formulation, at least with these choices of similarity and redundancy functions.

5.5 Adding compression to the ILP

Moving from extractive to abstractive summarization, while desirable in the long run, is a daunting challenge. Incorporating sentence compression into extractive summarization is something of a middle ground; while the modifications we can make to the source sentences are limited, we have an exponentially expanded set of textual units to choose from. This gives us improved flexibility for removing redundancy, for example. Here I'll describe how the ILP formulation for maximum coverage selection can be extended with additional constraints to include sentence compression.

One way to combine selection and compression is to generate compressed candidates for each sentence, creating an expanded pool of input sentences, and employ some redundancy removal on the final selection [Madnani *et al.*, 2007]. This approach can be adapted so that the ILP decides which alternatives to pick. If each compression candidate belongs to a group g_k corresponding to its source sentence, then we can include a constraint in the ILP to ensure that at most one sentence can be selected from each group (which also includes the source):

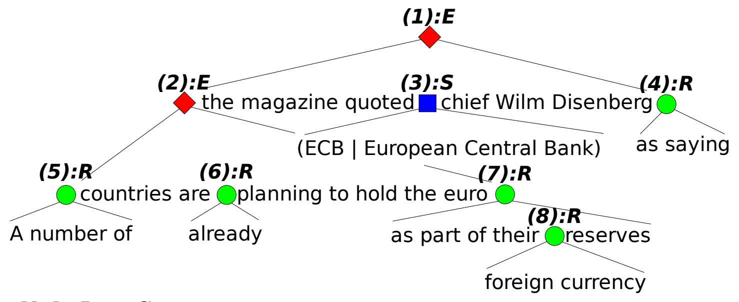
$$\sum_{i \in g_k} s_i \leq 1, \, \forall g_k$$

Assuming that all the compressed candidates are themselves well-formed, meaningful sentences, this approach should generate higher quality summaries. In general, however, compression algorithms can generate an exponential number of candidates.

Most compression strategies are based on the parse-tree structure of a sentence, where some subset of nodes are removed that preserve the grammaticality of the sentence [Knight and Marcu, 2000; McDonald, 2006; Turner and Charniak, 2005]. While we'll focus on node removals, we'll also allow extraction (a sub-sentence, perhaps the content of a quotation, may be used independently and the rest of the sentence is dropped) and substitution (one substring is replaced by another: US replaces United States, for example).

Arbitrary combinations of these operations are too general to be represented efficiently in an ILP. In particular, we need to compute the length of a sentence and the concepts it covers for all compression candidates. Thus, we insist that the operations only affect non-overlapping spans of text.

In our tree representation of a sentence, nodes correspond to compression operations and leaves map to words. Each node includes the word length it contributes to the sentence recursively, as the sum of the lengths of its children. Similarly, the concepts covered by a node are the union of the concepts covered by its children. When a node is active in the ILP, the words below it in the tree are included in the summary; active words contribute to the length constraint and objective score. Figure 5.3 gives an example of a tree representation, showing derivations of some compression candidates.



Node	Len.	Concepts
(1):E	6	{the_magazine, magazine_quoted, chief_Wilm, Wilm_Disenberg}
(2):E	7	{countries_are, planning_to, to_hold, hold_the, the_euro}
(3):S	0	{}
(3a)	1	{ECB}
(3b)	3	{European_Central, Central_Bank}
(4):R	2	{as_saying}
(5):R	3	{a_number, number_of}
(6):R	1	{}
(7):R	5	{as_part, part_of, reserves}
(8):R	2	{foreign_currency}

- Original: *A number of Countries are already planning to hold the euro as part of their foreign currency reserves, the magazine quoted European Central Bank chief Wim Duisenberg as saying.*
- [1,2,5,3a]: *A number of countries are planning to hold the euro, the magazine quoted ECB chief Wim Duisenberg.*
- [2,5,6,7,8]: *A number of countries are already planning to hold the euro as part of their foreign currency reserves.*
- [2,7,8]: *Countries are planning to hold the euro as part of their foreign currency reserves.*
- [2]: *Countries are planning to hold the euro.*

Figure 5.3: A compression tree for an example sentence. E-nodes (diamonds) can be extracted and used as independent sentences; R-nodes (circles) can be removed; S-nodes (squares) contain substitution alternatives. The table shows the word bigram concepts covered by each node and the length it contributes to the summary. Examples of resulting compression candidates are given on the right side, with the list of nodes activated in their derivations.

This framework can be used to implement a wide range of compression techniques. As a baseline proof-of-concept, we derive the compression tree from the sentence's parse tree given by the Berkeley parser [Petrov and Klein, 2007], and use a set of rules to label parse tree nodes with compression operations. For example, declarative clauses containing a subject and a verb are labeled with the extract (E) operation; adverbial clauses and non-mandatory prepositional clauses are labeled with the remove (R) operation; acronyms can be replaced by their full form by using substitution (S) operations, and a primitive form of co-reference resolution is used to allow the substitution of noun phrases with their referents.

While this compression framework gave a small improvement in ROUGE over the extractive system, Pyramid and Linguistic Quality scores declined significantly. An analysis of the resulting summaries showed that the rules used for identifying candidates for the compression operations fail to ensure that all compression candidates are valid sentences; about 60% of the summaries contained ungrammatical sentences.

Still, the ILP gave exact solutions quickly (1-3) seconds per problem; the model represents an efficient way to perform joint extraction and compression. My more recent work on joint modeling [Berg-Kirkpatrick *et al.*, 2011] adapts this framework for learning concept weights and weights for deleting parse tree nodes at the same time.

Chapter 6

Ordering Sentences

All the methods I've discussed so far involve choosing a set of sentences, neglecting the order in which they are presented. Since a great deal of meaning would be lost, or at least convoluted, if a document's sentences were presented in random order, we can expect the order of sentences in a summary to have some bearing on that summary's quality. For example, when Barzilay et al. asked people to order 8 sentences into a summary, 50 people created only 21 unique orderings, with significant overlap. While there are a variety of coherent orderings, this result suggests that the set of coherent orderings is a small subset of all possible orderings [Barzilay *et al.*, 2002].

6.1 Ordering is hard

A variety of work on sentence ordering has produced only relatively small improvement over a chronological baseline (sentences are ordered by source document publication date first, and second by their order of appearance in the source). The primary competing feature is some form of *lexical cohesion*, as studies have shown that poor readability tends to arise from jumps between topics.

Conroy et al., for example, compute the sentence similarity (some variant of the usual cosine distance) between all pairs of selected sentences, and then choose the ordering that minimizes the sum of all pairwise distances [Conroy *et al.*, 2006]. The resulting Traveling Salesman problem [Lin, 1965] is NP-hard and thus exponential in the number of sentences, but many good approximations exist, and for short summaries, exact solutions are feasible. While such a straightforward optimization is appealing, the connection between sentence similarity and actual semantic cohesion is tenuous.

Bollegala et al., attempt to combine such lexical cohesion metrics with chronological features in a classifier with some success [Bollegala *et al.*, 2010]. Still, the chronological information proved by far the strongest individual feature, and the classifier gave significant but small improvement.

Certainly, current methods could stand to improve on the summary re-ordering problem. But perhaps a more serious issue, especially when the desired summary length is only a few sentences (as in the 2008 and 2009 datasets), is choosing a set of sentences that *have the potential* to be ordered in a reasonable fashion. One possibility is to try solving the ordering problem jointly with the selection problem. A simpler, practical solution is instead to try selecting *independent* sentences, sentences that can stand alone without relying on other context. If we can select independent sentences, perhaps we can bypass the pesky ordering problem.

6.2 Independent sentences

The focus on chronology and lexical cohesion obscures the biggest obstacle to coherence in an extractive summary: coreference. A person or organization might be introduced with identifying detail in one sentence, but subsequent references are often pronominal. "He refused to comment for the story" has little meaning without sentences that give context for "he" and "the story".

One way to fix incoherent summaries is to model the coreference structure in the source documents, requiring, for example, that a sentence can only be included in a summary if its dependencies are also included. Certainly, cross-document coreference resolution is an active area of research [Haghighi and Klein, 2010; Rahman and Ng, 2011], but even the best systems give 65%-75% accuracy on sets of news documents like ours.

Before going down this road, though, it would be prudent to see if the set of sentences that has dependencies are really important. Here, I'll describe a classification approach that takes a pronoun and features of its containing sentence, and outputs a decision: resolved or unresolved. As it turns out, the classifier performs at over 90% accuracy, so we can tell fairly reliably if a sentence contains pronouns that require other sentences for context. Removing these non-independent sentences ought to improve overall coherence, and thus overall quality so long as content is not lost.

The classifier is trained from the coreference resolution annotations in OntoNotes 2.9 [Hovy *et al.*, 2006]. A pronoun is considered to be resolved if a non-pronominal reference to the same entity is present in the sentence. The processing pipeline for removing sentences with unresolved pronouns is as follows:

1. Parse an input sentence using the Berkeley constituency parser [Petrov and Klein, 2007].

2. Locate potential pronouns with the "PRP" and "PRP$" part-of-speech tags.

3. Extract features from the parse tree: pronoun position, relative position of NPs, other pronouns in the sentence, etc.

4. Prune the sentence if at least one unresolved pronoun is identified.

We train a linear classifier, in this case AdaBoost [Schapire, 1999] to help make the learned weights interpretable[1]. 6,400 pronoun instances, of which 50% are positive examples (unresolved), are used as training data. Table 6.1 shows the first few (most important) features selected by the classifier along with performance on a 1000-example test set.

The most useful piece of information is whether there is an noun phrase appearing before the pronoun. If there is, as in "President Obama said he condemned the attacks", usually this NP directly resolves the pronoun. In fact, all of the cases (in the test set) of unresolved pronouns have no preceding NP; of course, some of these find resolution later in the sentence, so more features are needed. Top performance on a held-out set of 9% error is reached with around 50 features.

Feature (relative to pronoun)	**Error**	**Recall**	**Precision**
No NP before	34%	100%	50%
Closest preceding NP starts with NNP	22%	90%	68%
Closest preceding NP starts with PRP	19%	75%	88%
Previous word is <COMMA>	17%	79%	87%
Previous word is <QUOTES>	15%	82%	87%

Table 6.1: Top five features selected by AdaBoost for classification of pronouns as resolved or unresolved.

Table 6.2 shows the effect of pruning sentences with a variety of decision thresholds. The results suggest that 30% of sentences can be pruned safely, perhaps even improving ROUGE scores. Also note that this pruning is fairly independent of pruning derived from a content classifier, as described in 4.5.1. Together, the two classifiers can safely prune over 50% of the input sentences, giving a marginal improvement in ROUGE. While I have not conducted an evaluation of the linguistic quality of the resulting summaries, they certainly contain fewer unresolved pronouns. And linguistic quality scores in the 2009 evaluation were comparable to the best (non-baseline) system (see Figure 4.3).

This classification approach adequately handles pronoun coreference cases, but still many context-dependent sentences remain. The problem is that coreference is more complex than names and pronouns: "The explosion", "Obama's decision", and

[1]We use an open-source implementation of AdaBoost [Favre *et al.*,].

	2008		2009	
Threshold	**Pruned**	**ROUGE-2**	**Pruned**	**ROUGE-2**
1	0%	11.54	0%	11.92
0.1	29%	11.66	30%	12.04
0.01	37%	11.59	37%	11.80
0.001	44%	11.64	45%	11.53
0	54%	11.39	53%	11.31

Table 6.2: ROUGE results for different sentence-pruning thresholds with the unresolved pronoun classifier.

"consequences", for example, all require elaboration. Further improvements in linguistic quality will likely involve more sophisticated identification of context-dependent phrases. Whether it will be important to map the dependency structure remains to be seen.

Chapter 7

Evaluating Summaries

7.1 N-gram recall

ROUGE, as reported throughout this thesis, has been the object of considerable criticism. As I understand the calculation reported in Lin's original paper, ROUGE-1 is computed by:

$$\frac{\displaystyle\sum_{S \in Ref} \sum_{w \in S} \max(C_{candidate}(w), C_{ref}(w))}{\displaystyle\sum_{S \in Ref} \sum_{w \in S} C_{ref}(w)} \tag{7.1}$$

Where Ref is the set of reference summaries (there are four of these for each problem in our datasets), and w is a word in the set of words in S. $C_{candidate}(w)$ is the number of times that word w appears in the candidate summary and $C_{ref}(w)$ is the number of times that word w appears in the references. ROUGE-n replaces w with an n-gram.

This is an odd calculation for a number of reasons, but one obvious complaint, especially if you subscribe the maximum coverage model for summarization, is that repeated n-grams in the candidate summary are rewarded. To fix this issue, and to simplify the algorithm for automatic evaluation, I propose n-gram recall as a new metric. With $G(y)$ the set of n-grams in the candidate summary and $G(y^*)$ the set of n-grams in the reference summaries, n-gram recall is defined as:

$$\frac{|G(Y) \cap G(Y^*)|}{|G(Y^*)|} \tag{7.2}$$

Table 7.1 compares correlations to show that nothing is lost by simplifying ROUGE. Alternate correlation metrics show similar results.

	2008 Correlation		**2009 Correlation**	
Metric	**OQ**	**Pyr**	**OQ**	**Pyr**
ROUGE-1	0.85	0.91	0.81	0.89
Recall-1	0.87	0.92	0.84	0.91
ROUGE-2	0.89	0.94	0.88	0.97
Recall-2	0.90	0.94	0.89	0.97
ROUGE-3	0.91	0.94	0.88	0.97
Recall-3	0.92	0.94	0.89	0.97
ROUGE-4	0.91	0.92	0.86	0.94
Recall-4	0.92	0.92	0.87	0.94

Table 7.1: Pearson correlations between systems' average manual scores (Overall Quality and Pyramid) and average automatic scores (classic ROUGE and the new simple recall metric).

Note that unlike ROUGE, n-gram recall throws all the reference n-grams together in a single set, ignoring the fact that some of those n-grams appear in all the references and others appear in only one, for example. Intuitively, as we saw with weighting n-grams in the source documents, the more references a word appears in, the important it is likely to be. Pyramid takes this point of view, giving weight to a fact equal to the number of references that include it. But experiments show that applying this intuition to n-grams is not so straightforward.

Rather than count the total number of overlapping n-grams, let's subdivide these so we count the number of overlapping n-grams that appear in 1 reference, 2 references, 3, and 4. Table 7.2 shows Pearson correlations for the counts in these subdivisions. Unigrams show the expected behavior: the number of overlaps you have with words appearing in all the references is more indicative of quality than the number of overlaps you have with words appearing in a single reference. But higher order n-grams have the opposite behavior. Since 4-grams are the most extreme in this regard, it seems reasonable to assume that part of the problem is sparsity. Almost no 4-grams appear in all 4 reference summaries, so using this statistic is unreliable.

There is still another hidden source of variability here. Included in the set of reference words are content words like "president" and "earthquake", but also function words like "the" and "of", which are presumably much less important. N-gram recall makes no distinction between different kinds of words; in fact, removing stopwords (with a list) hurts correlation both with ROUGE and n-gram recall. So another reason that n-grams appearing many times are potentially less useful is because these are more likely to have more function words and fewer content words.

Thus one way to pursue improved automatic evaluation metrics is to find a way

	Number of references			
N	**1**	**2**	**3**	**4**
1	0.77	0.81	0.84	0.87
2	0.89	0.86	0.87	0.86
3	0.89	0.83	0.83	0.80
4	0.87	0.78	0.79	0.46

Table 7.2: Pearson correlations between systems' average Overall Quality scores and average automatic scores broken down by order n and number of references; 2009 data.

to weight n-grams in the references according to their predictive potential.

7.2 Maximum recall

Another reason to prefer n-gram recall to ROUGE is that it lends itself more naturally to optimization. In particular, finding the maximum-ROUGE summary given a set of references is difficult because of the non-linearity in the max; but finding the maximum recall summary is a straightforward application of the maximum coverage problem described earlier. In place of the values estimated from the input documents, we can substitute an indicator for whether the n-gram appears in the references. The ILP gives efficient solutions.

Figure 7.1 is the result of a manual evaluation of a variety of summaries: two baseline systems (B1 and B2), our state-of-the-art (SOA) system, the maximum recall oracle (MRO) summaries, and the human-written abstracts (H). It appears that even though the oracle's ROUGE scores are high, they are still roughly correlated with quality; human summaries, on the other hand, which tend to use different language, occupy a separate space in the plot. It seems reasonable to conclude that (1) human summaries and extractive summaries exhibit different relationships between ROUGE and quality, and (2) even at high values, ROUGE continues to correlate with quality, though the relationship is non-linear: improvements in ROUGE score mean less at larger values.

The evaluation showed that the maximum recall extracts were of significantly better quality than the best automatic extractive system. Further reinforcing the point that systems have not achieved the best possible results in extraction, Genest et al. [Genest *et al.*, 2010] conducted an evaluation of manual extractive summaries using the 2009 dataset. They found that the manual extracts were comparable in Pyramid score to the best automatic system, but significantly worse in linguistic quality and overall quality.

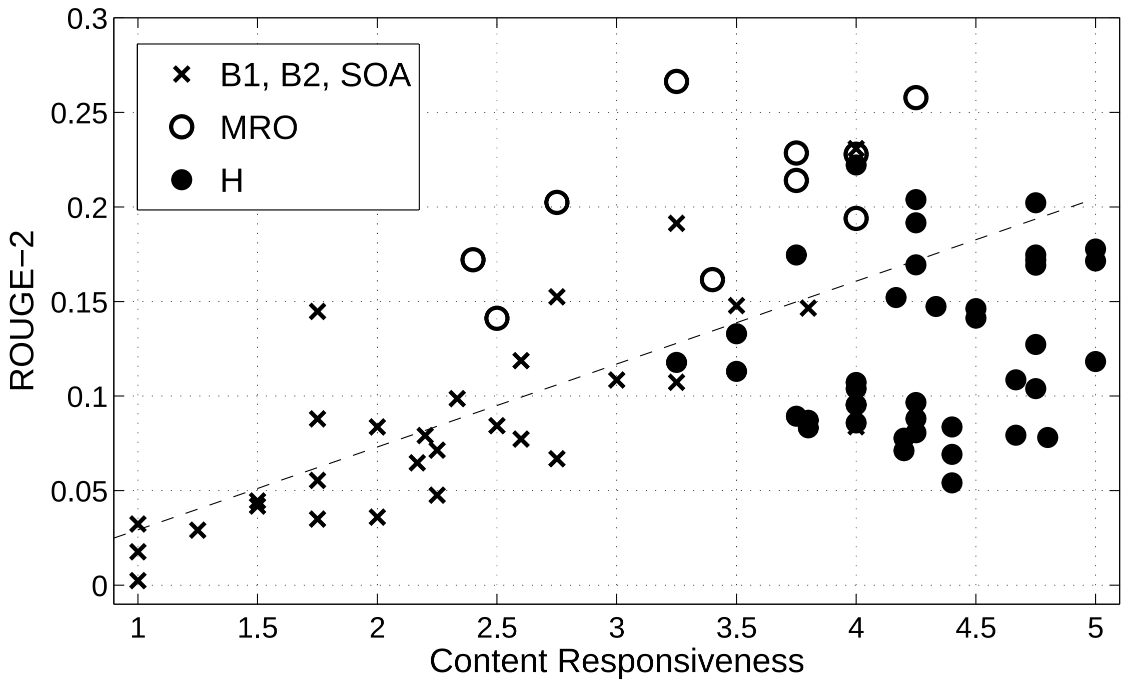


Figure 7.1: ROUGE-2 vs. manually evaluated quality is plotted for each summary. The automatic systems B1, B2, and State of the art (SOA) are grouped together and the trend line is a linear fit for this data. The Maximum Recall Oracle (MRO) results all appear above the trend line, while the Human (H) results seem uncorrelated.

7.3 Crowdsourcing

Manual evaluation is time-consuming. Ideally, a judge would read the original set of documents before deciding how well the important aspects are conveyed by a summary. A typical 10-document problem could reasonably involve 25 minutes of reading or skimming and 5 more minutes for assessing a 100-word summary. Since summary output can be quite variable, at least 30 topics should be evaluated to get a robust estimate of performance. Assuming a single judge evaluates all summaries for a topic (more redundancy would be better), we get a rough time estimate: 17.5 hours to evaluate two systems.

Thus it is of great interest to find ways of speeding up evaluation while minimizing subjectivity. Amazon's Mechanical Turk (MTurk) system has been used for a variety of labeling and annotation tasks [Snow *et al.*, 2008], but such crowd-sourcing has not been tested for summarization.

Here I describe an experiment to test whether MTurk is able to reproduce system-level rankings that match expert opinion [Gillick and Liu, 2010]. Unlike the results of other crowd-sourcing annotations for natural language tasks, here it seems that non-expert judges are unable to provide expert-like scores and tend to disagree significantly with each other.

7.3.1 Agreement and consistency

In the official 2009 evaluation, each summary was judged by one of eight experts for "Overall Quality" and "Linguistic Quality" on a 1 ("very poor") to 10 ("very good") scale. Unfortunately, the lack of redundant judgments means we cannot estimate inter-annotator agreement. However, we note that out of all 4576 submitted summaries, there are 226 pairs that are identical, which allows us to estimate annotator consistency. Table 7.3 shows that an expert annotator will give the same summary the same score just over half the time.

	Score Difference				
	0	1	2	3	mean
OQ	119	92	15	0	0.54
LQ	117	82	20	7	0.63

Table 7.3: Identical summaries often were given different scores by the same expert human judge (2009 data). Counts of absolute score differences are shown for Overall Quality (OQ) and Linguistic Quality (LQ).

7.3.2 Designing a task for non-experts

One way to dramatically speed up evaluation is to use the experts' reference summaries as a gold standard, leaving the source documents out entirely. This is the idea behind automatic evaluation with ROUGE, which measures n-gram overlap with the references, and assisted evaluation with Pyramid, which measures overlap of facts or "Semantic Content Units" with the references. The same idea has also been employed in various manual evaluations, for example by Haghighi and Vanderwende [2009], to directly compare the summaries of two different systems. The potential bias introduced by such abbreviated evaluation has not been explored.

The overall structure of the Human Intelligence Task (HIT) we designed for summary evaluation is as follows: The worker is asked to read the topic and description, and then two reference summaries (there is no mention of the source documents). The candidate summary appears next, followed by instructions to provide scores between

1 (very poor) and 10 (very good) in each category[1]. Mouse-over on the category names provides extra details, copied with slight modifications from Dang [2007].

The initial HIT design asked workers to perform a head-to-head comparison of two candidate summaries, but we found this unsatisfactory for a number of reasons. First, many of the resulting scores did not obey the transitive property: given summaries x, y, and z, a single worker showed a preference for $y > x$ and $z > y$, but also $x > z$. Second, while this kind of head-to-head evaluation may be useful for system development, we are specifically interested here in comparing non-expert MTurk evaluation with expert TAC evaluation.

We went through a few rounds of revisions to the language in the HIT after observing worker feedback. Specifically, we found it was important to emphasize that a good summary not only responds to the topic and description, but also conveys the information in the references.

Only workers with at least a 96% HIT approval rating[2] were allowed access to this task. We monitored results manually and blocked workers (rejecting their work) if they completed a HIT in under 25 seconds. Such suspect work typically showed uniform scores (usually all 10s). Nearly 30% of HITs were rejected for this reason.

To encourage careful work, we included this note in our HITs: "High annotator consistency is important. If the scores you provide deviate from the average scores of other annotators on the same HIT, your work will be rejected. We will award bonuses for particularly good work." We gave a few small bonuses ($0.50) to workers who left thoughtful comments.

We experimented with a few different compensation levels and observed a somewhat counter-intuitive result. Higher compensation ($.10 per HIT) yielded lower quality work than lower compensation ($.07 per HIT), judging by the number of HITs we rejected. It seems that lower compensation attracts workers who are less interested in making money, and thus willing to spend more time and effort. There is a trade-off, though, as there are fewer workers willing to do the task for less money.

7.3.3 Experiments and analysis

To assess how well MTurk workers are able to emulate the work of expert judges employed in the official evaluation, we chose a subset of systems and analyze the results of the two evaluations. The systems were chosen to represent the entire range of average Overall Quality scores. System F is a simple lead baseline, which generates a summary by selecting the first sentences up to 100 words of the most recent document. The rest of the systems were submitted by various track participants. The MTurk

[1]Besides Overall Quality and Linguistic Quality, we include Information Content, to encourage judges to distinguish between content and readability.

[2]MTurk approval ratings calculated as the fraction of HITs approved by requesters.

evaluation included two-times redundancy. That is, each summary was evaluated by two different people. The cost for the full evaluation, including 44 topics, 8 systems, and $2x$ redundancy, at \$.07 per HIT, plus 10% commission for Amazon, was \$55.

Table 7.4 shows average scores for the two evaluations. The data suggest that the MTurk judges are better at evaluating Linguistic Quality than Content or Overall Quality. In particular, the MTurk judges appear to have difficulty distinguishing Linguistic Quality from Content. We will defend these claims with more analysis, below.

System	**Official**		**MTurk**		
	OQ	**LQ**	**OQ**	**LQ**	**C**
A	5.16	5.64	7.03	7.27	7.27
B	4.84	5.27	6.78	6.97	6.78
C	4.50	4.93	6.51	6.85	6.49
D	4.20	4.09	6.15	6.59	6.50
E	3.91	4.70	6.19	6.54	6.58
F	3.64	6.70	7.06	7.78	6.56
G	3.57	3.43	5.82	6.33	6.28
H	3.20	5.23	5.75	6.06	5.62

Table 7.4: Comparison of Overall Quality (OQ) and Linguistic Quality (LQ) scores between the official and MTurk evaluations. Content (C) is evaluated by MTurk workers as well. Note that system F is the lead baseline.

7.3.4 Worker variability

The first important question to address involves the consistency of the workers. We cannot compare agreement between the evaluations, but the MTurk agreement statistics suggest considerable variability. In Overall Quality, the mean score difference between two workers for the same HIT is 2.4 (the standard deviation is 2.0). The mean is 2.2 for Linguistic Quality (the standard deviation is 1.5).

In addition, the expert judges show more similarity with each other—as if they are roughly in agreement about what makes a good summary. We compute each judge's average score and look at the standard deviation of these averages for the two groups. The expert standard deviation is 1.0 (ranging from 3.0 to 6.1), whereas the MTurk standard deviation is 2.3 (ranging from 1.0 to 9.5). Note that the average number of HITs performed by each MTurk worker was just over 5.

Finally, we can use regression analysis to show what fraction of the total score variance is captured by judges, topics, and systems. We fit linear models in **R** using binary indicators for each judge, topic, and system. Redundant evaluations in the MTurk set are removed for unbiased comparison with the official set. Table 7.5 shows that the differences between the evaluations are quite striking: Taking the official data alone, the topics are the major source of variance, whereas the judges are the major source of variance in the MTurk data. The systems account for only a small fraction of the variance in the MTurk evaluation, which makes system ranking more difficult.

Eval	**Judges**	**Topics**	**Systems**
Official	0.28	0.40	0.13
MTurk	0.44	0.13	0.05

Table 7.5: Linear regression is used to model Overall Quality scores as a function of judges, topics, and systems, respectively, for each data set. The R^2 values, which give the fraction of variance explained by each of the six models, are shown.

7.3.5 Ranking comparisons

The official evaluation, while lacking redundant judgments, was a balanced experiment. That is, each judge scored every system for a single topic. The same is not true for the MTurk evaluation, and as a result, the average per-system scores shown in Table 7.4 may be biased. As a result, and because we need to test multiple system-level differences simultaneously, a simple t-test is not quite sufficient. We use Tukey's Honestly Significant Differences (HSD), explained in detail by Yandell [1997], to assess statistical significance.

Tukey's HSD test computes significance intervals based on the range of the sample means rather than individual differences, and includes an adjustment to correct for imbalanced experimental designs. The **R** implementation takes as input a linear model, so we model scores using binary indicators for (J)udges, (T)opics, and (S)ystems (see equation 7.3), and measure significance in the differences between system coefficients (δ_k).

$$score = \alpha + \sum_i \beta_i J_i + \sum_j \gamma_j T_j + \sum_k \delta_k S_k \tag{7.3}$$

Table 7.6 shows system rankings for the two evaluations. The most obvious discrepancy between the official and MTurk rankings is system F, the baseline. Both expert and MTurk judges gave F the highest scores for Linguistic Quality, a reasonable result given its construction, whereas the other summaries tend to pull sentences

Eval	**Ranking**							
Official (OQ)	A	B	C	D^A	E^B	F^C	G^C	H^D
MTurk (OQ)	F	A	B	C	E^F	G^F	D^B	H^B
TAC (LQ)	F	A^F	B^F	H^F	C^F	E^A	D^B	G^E
MTurk (LQ)	F	A	B^F	C^F	D^F	E^F	H^C	G^C
MTurk (C)	A	B	E	F	D	C	G^A	H^D

Table 7.6: Systems are shown in rank order from highest (left) to lowest (right) for each scoring metric: Overall Quality (OQ), Linguistic Quality (LQ), and Content (C). The superscripts indicate the rightmost system that is significantly different (at 95% confidence) according to Tukey's HSD test.

out of context. But the MTurk judges also gave F the highest scores in Overall Quality, suggesting that readability is more important to amateur judges than experts, or at least easier to identify. Content appears the most difficult category for the MTurk judges, as few significant score differences emerge. Even with more redundancy, it seems unlikely that MTurk judges could produce a ranking resembling the official Overall Quality ranking using this evaluation framework.

What does this mean for future evaluations? If we want to assess overall summary quality—that is, balancing content and linguistic quality like expert judges do—we will need to redesign the task for non-experts. Perhaps MTurk workers will be better able to understand Pyramid evaluation, which is designed to isolate content. Extrinsic evaluation, where judges use the summary to answer questions derived from the source documents or the references, as done by Callison-Burch for evaluation of Machine Translation systems [2009], is another possibility.

Finally, these results suggest that anyone conducting an evaluation of summarization systems using non-experts should calibrate their results by asking their judges to score summaries that have already been evaluated by experts.

7.4 Towards extrinsic evaluation

So far, all the evaluation methods discussed in this chapter have been *intrinsic*. That is, they are based on the idea of quality without regard to any particular task. While some might consider an elegant summary an end in of itself, most potential summary users are more pragmatic, so the value of a summary should be a measure of how useful it is for performing some task.

Towards this end, a few research efforts have attempted to perform this kind of task-based *extrinsic* evaluation. Early results have been somewhat encouraging. Mckeown et al. [McKeown *et al.*, 2005] compared the performance of people trying to gather facts about a set of documents. Different subsets of people were given the original documents, headlines only, automatic summaries, and human-generated abstracts. The study showed good evidence that people equipped with summaries gave better answers than people without summaries. Further, the higher the quality of the summary, the more satisfying the user experience. Dorr et al. [Dorr *et al.*, 2005] conducted a related study, finding for example, that a summary allowed users to complete a task 65% faster than a (single) original document.

While these studies help validate summarization as a useful area of research and suggest that non-trivial progress has been made, I am not convinced that a shift towards more extrinsic evaluation is necessary. By nature, extrinsic evaluation is certainly more practical, but by necessity, less general. Since extrinsic metrics have tended to correlate well with intrinsic metrics, validating academic work on summarization with intrinsic evaluation seems reasonable. Occasional extrinsic evaluations of the best performing systems are a good way to validate progress, and useful for companies trying to benchmark performance on more specific tasks.

Chapter 8

In Summary

8.1 Review

The previous chapters have addressed the range of problems facing the designer of an automatic summarization system.

Input documents need to be cleaned and segmented into useful units. Toward this end, I described a set of features that are able to distinguish sentence-ending periods from non-sentence-ending periods that follow abbreviations. Feature weights are learned from data and can be used for discrimination at very low error rates ($<$0.5%) on a variety of corpora.

There are many methods for assigning value to sentences and then assembling a summary. I described Maximum Marginal Relevance and SumBasic because they begin to chart a research trajectory that moved away from Information Retrieval and towards functional criteria for scoring a summary based on statistics from the set of input documents. The particular objective function I used, called maximum coverage, implements the idea that a summary is worth the sum the of the values of the concepts it covers. Even a very rough approximation of the notion of a concept with a word bigram gives state-of-the-art results.

Maximum coverage also has some nice theoretical properties. It lends itself readily to an Integer Linear Program formulation that allows for fast exact solutions; it is also submodular, implying that greedy approximations are guaranteed to come reasonably close to optimal solutions. Extending the framework to include syntactic sentence compression is fairly straightforward.

One way to approach the tricky issue of sentence ordering is to prune sentences that make ordering difficult. The primary culprit here is pronoun reference. While coreference resolution is difficult, I show that identifying pronouns that are not resolved locally (in the current sentence) is much easier. Using such a classifier to prune 20-40% of sentences before selection can even improve ROUGE.

Lastly, I addressed a few issues in system evaluation. I showed that a simpler version of ROUGE, called n-gram recall, works at least as well (in correlation with manual metrics). I also showed directly how subjective evaluation can be: non-expert readers tended not to be able to reproduce expert judgments of content as they appeared distracted by linguistic quality.

8.2 Data

The data used in my thesis work comes from the TAC and DUC workshops. But there are a variety of other data sets available that may be worth pursuing, especially because the TAC/DUC sets are quite small. I have found that machine learning, especially structured prediction, tends to be unstable with such small training and test sets. For example, feature engineering is difficult when changing the feature set has a less significant effect than simply adding minimal random noise during training.

Fortunately, the LDC recently released a large corpus of New York Times articles, spanning nearly 20 years (1988-2007). About 650,000 of these have accompanying summaries (mostly abstractive). Even after fairly aggressive pruning to remove short documents, or documents with short abstracts, as well as corrections and editorials, nearly 20% of these, over 100,000 pairs remain. A few example abstracts are shown in Table 8.1. Each document also comes labeled with a variety of topic tags and identifying keywords. Though the corpus consists of only single-document summaries, its strength is its size, which begs the question: What can be learned from a large collection of documents and summaries?

The feature values we learned with regression (see Table 4.4) can be estimated from just a few documents; with such a simple model, thousands of training examples are not really necessary. Perhaps we can learn feature weights conditional on the document type (sports, business, etc.). Perhaps we can estimate the probability of each input word appearing in the summary (see Table 2.1), the first step towards a kind of summary model, akin to the acoustic model in speech recognition or the translation model in machine translation.

In addition to news articles and their various summaries, there are collections of research papers and abstracts, including the PubMed database and the ACL anthology. Legal documents often include hand-written summaries. And, all sorts of movie, music, book, and product reviews are often summarized by a star rating, perhaps a more appealing unit of prediction than an actual summary.

Rising Costs Put M.T.A. Projects At Risk of Delay

Metropolitan Transportation Authority faces rising costs that could imperil or delay subway renovations, maintenance of signal systems and purchase of New York City buses and subway cars; is halfway through five-year, $21 billion program to expand and improve transit system one estimate has program $1.4 billion over budget; major problem has been rapid increase in New York City construction costs, as result of increasing materials costs and large number of new projects, which gives contractors leverage to charge more; some transit experts are concerned that smaller repair and maintenance projects may be sacrificed in order to go ahead with high-profile projects; authority now faces two-pronged problem of its long-term spending plan and its day-to-day operating budget, which is expected to have large operating deficits beginning in 2008.

Stigma Aside, Wall St. Finds A Lot to Like About Tobacco

Investors react exuberantly to news that Altria Group, company once known as Philip Morris, will spin off its Kraft Foods division and become primarily tobacco company; Kraft Foods has stagnated in rapidly changing marketplace; cigarettes have advantages over other consumer products: are addictive, inexpensive to make and there is global market for them; future prospects for cigarette makers are very attractive in developing countries, where smoking has not declined; recent court decisions favorable to Altria have caused its stock to soar; some health advocates say that dwindling social acceptability of cigarettes will hurt long-term prospects of companies like Altria; analysts question future of independent Kraft Foods, which faces formidable challenges; lawyer has threatened to file injunction to stop divestiture but several analysts dismiss chances of injunction's success; charts of Altria Group's stock prices over decade.

Figure 8.1: Sample abstracts (with titles) from the NYT corpus. These have been cleaned up somewhat, and selected from a subset of the data with very short abstracts and documents removed.

8.3 Methods

Maximum coverage appears to be a good general objective for summarization. That is, if we only had a reasonable representation of facts and good estimates of their relative values, this objective would yield meaningful summaries.

I have found that limiting the representation of facts to word n-grams makes the estimation of their values quite difficult. It is hard to beat a frequency-based approach, even with fairly advanced techniques. This is because n-grams provide only noisy fragments of facts: "he wrote" or "decided not", for example, are faint echoes of actual facts in the documents and their repetition across documents vaguely suggest the importance of those underlying facts. Frequency-based valuation also struggles in cases like "George W. Bush”: bigrams "George W.” and "W. Bush" both get the same counts, effectively multiplying the value of the full name by two. Under these circumstances, it is hard to expect any learning system to tease apart relative weights of counts, order, and so on.

So while it is conceivable that better estimation of n-gram values could improve summaries, improving the representation of facts is likely to be more productive. For example, we know that human-written abstract sentences tend to have more verbs than the original documents. Perhaps verb-based templates, like subject-verb-object, would be more meaningful. Of course, one benefit of n-grams is that counting them is easy; I used word-stemming to limit diversity. Counting the number of occurrences of a particular subject-verb-object would be more difficult, probably requiring cross-document coreference resolution of nouns and pronouns.

Thus my suggestion for future research is to focus on the extraction of facts from documents, ideally across very large document collections like the New York Times corpus. To start, applying a few rules to parser output can give subject-verb-object triples, and approximate string matching can help with robustness. Note that we cannot expect a summary built from non-n-gram facts to outperform one built from n-gram facts as measured by ROUGE or n-gram recall. In fact, when working on the problems of identifying and matching facts, full summarization is a bit of a distraction. These problems are useful in of themselves and deserve their own treatment. Better summarization systems will follow eventually.

Bibliography

[Aberdeen *et al.*, 1995] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. MITRE: description of the Alembic system used for MUC-6. In *Proceedings of the 6th conference on Message understanding*, pages 141–155. Association for Computational Linguistics Morristown, NJ, USA, 1995.

[Barzilay *et al.*, 2002] R. Barzilay, N. Elhadad, and K.R. McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.

[Berg-Kirkpatrick *et al.*, 2011] T. Berg-Kirkpatrick, D. Gillick, and D. Klein. Learning to jointly extract and compress. Association for Computation Linguistics, 2011.

[Blei *et al.*, 2003] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[Bollegala *et al.*, 2010] D. Bollegala, N. Okazaki, and M. Ishizuka. A bottom-up approach to sentence ordering for multi-document summarization. *Information Processing & Management*, 46(1):89–109, 2010.

[Callison-Burch and Baltimore, 2009] C. Callison-Burch and M. Baltimore. Fast, Cheap, and Creative: Evaluating Translation Quality Using AmazonÃŢs Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, 2009.

[Carbonell and Goldstein, 1998] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. *Research and Development in Information Retrieval*, pages 335–336, 1998.

[Celikyilmaz and Hakkani-Tur, 2010] A. Celikyilmaz and D. Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824, 2010.

[Conroy and O'Leary, 2001] J.M. Conroy and D.P. O'Leary. Text Summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page 407, 2001.

[Conroy *et al.*, 2006] J.M. Conroy, J.D. Schlesinger, D.P. O'Leary, and J. Goldstein. Back to basics: Classy 2006. In *Proceedings of DUC*, volume 6, 2006.

[Dang, 2007] Hoa Trang Dang. Overview of DUC 2007. In *Proceedings of DUC'07 workshop*, pages 1–10, 2007.

[Daumé III and Marcu, 2005] H. Daumé III and D. Marcu. Induction of word and phrase alignments for automatic document summarization. *Computational Linguistics*, 31(4):505–530, 2005.

[Dorr *et al.*, 2005] B.J. Dorr, C. Monz, R. Schwartz, and D. Zajic. A methodology for extrinsic evaluation of text summarization: does ROUGE correlate? *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, page 1, 2005.

[Edmundson, 1964] HP Edmundson. Problems in automatic abstracting. *Communications of the ACM*, 7(4):263, 1964.

[Erkan and Radev, 2004] G. Erkan and D.R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(2004):457–479, 2004.

[Favre *et al.*,] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet. Icsiboost. http://code.google.come/p/icsiboost.

[Filatova and Hatzivassiloglou, 2004] E. Filatova and V. Hatzivassiloglou. Event-based extractive summarization. In *Proceedings of ACL Workshop on Summarization*, volume 111, 2004.

[Gallo *et al.*, 1980] G. Gallo, PL Hammer, and B. Simeone. Quadratic knapsack problems. *Mathematical Programming Study*, 12:132–149, 1980.

[Genest *et al.*, 2010] P.E. Genest, G. Lapalme, and M. Yousfi-Monod. Hextac: the creation of a manual extractive run. In *Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA: National Institute of Standards and Technology*, 2010.

[Gillick and Favre, 2009] D. Gillick and B. Favre. A Scalable Global Model for Summarization. In *Proceedings of NAACL Workshop on Integer Linear Programming for Natural Language Processing*, 2009.

[Gillick and Liu, 2010] D. Gillick and Y. Liu. Non-expert evaluation of summarization systems is risky. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 148–151. Association for Computational Linguistics, 2010.

[Gillick *et al.*, 2010] D. Gillick, B. Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie. The icsi/utd summarization system at tac 2009. In *Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA: National Institute of Standards and Technology*, 2010.

[Gillick, 2009] D. Gillick. Sentence Boundary Detection and the Problem with the U.S. In *Proceedings of NAACL: Short Papers*, 2009.

[Goldstein *et al.*, 1999] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128. ACM, 1999.

[Goldstein *et al.*, 2000] Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, pages 40–48, 2000.

[Gong and Liu, 2001] Y. Gong and X. Liu. 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*, pages 19–25. ACM New York, NY, USA, 2001.

[Graff *et al.*, 2007] D. Graff, J. Kong, K. Chen, and K. Maeda. English gigaword third edition. *Linguistic Data Consortium*, 2007.

[Haghighi and Klein, 2010] A. Haghighi and D. Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393. Association for Computational Linguistics, 2010.

[Haghighi and Vanderwende, 2009] A. Haghighi and L. Vanderwende. Exploring Content Models for Multi-Document Summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, 2009.

[Hennig and Labor, 2009] L. Hennig and DAI Labor. Topic-based multi-document summarization with probabilistic latent semantic analysis. In *International Conference on Recent Advances in Natural Language Processing (RANLP)*, 2009.

[Hochbaum, 1996] D.S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. *PWS Publishing Co. Boston, MA, USA*, pages 94–143, 1996.

[Hofmann, 1999] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[Hovy *et al.*, 2006] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60. Association for Computational Linguistics, 2006.

[Jing and McKeown, 1999] H. Jing and K.R. McKeown. The decomposition of human-written summary sentences. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136. ACM, 1999.

[Kiss and Strunk, 2006] T. Kiss and J. Strunk. Unsupervised Multilingual Sentence Boundary Detection. *Computational Linguistics*, 32(4):485–525, 2006.

[Knight and Marcu, 2000] K. Knight and D. Marcu. Statistics-Based Summarization-Step One: Sentence Compression. In *Proceedings of the National Conference on Artificial Intelligence*, pages 703–710. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.

[Knight, 1999] K. Knight. A statistical MT tutorial workbook. In *The JHU Summer Workshop*, 1999.

[Krause and Guestrin, 2005] A. Krause and C. Guestrin. A Note on the Budgeted Maximization of Submodular Functions. Technical Report CMU-CALD-05, 103, Carnegie Mellon University, 2005.

[Kupiec *et al.*, 1995] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, 1995.

[Li *et al.*, 2007] S. Li, Y. Ouyang, W. Wang, and B. Sun. Multi-document summarization using support vector regression. In *Proceedings of DUC*, 2007.

[Lin and Bilmes, 2010] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics, 2010.

[Lin, 1965] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10):2245–2269, 1965.

[Lin, 2003] C.Y. Lin. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 1–8. Association for Computational Linguistics, 2003.

[Lin, 2004] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 25–26, 2004.

[Loper and Bird, 2002] E. Loper and S. Bird. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, page 70. Association for Computational Linguistics, 2002.

[Lovász, 1983] L Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art*, pages 235–257, 1983.

[Luhn, 1958] H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[Madnani *et al.*, 2007] N. Madnani, D. Zajic, B. Dorr, N.F. Ayan, and J. Lin. Multiple Alternative Sentence Compressions for Automatic Text Summarization. In *Proceedings of the 2007 Document Understanding Conference (DUC-2007) at NLT/-NAACL*, 2007.

[Marcus *et al.*, 1993] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):330, 1993.

[Martins and Smith, 2009] A.F.T. Martins and N.A. Smith. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 1–9. Association for Computational Linguistics, 2009.

[McDonald, 2006] R. McDonald. Discriminative sentence compression with soft syntactic constraints. In *Proceedings of the 11th EACL*, pages 297–304, 2006.

[McDonald, 2007] R. McDonald. A Study of Global Inference Algorithms in Multi-document Summarization. *Lecture Notes in Computer Science*, 4425:557, 2007.

[McKeown *et al.*, 2005] K. McKeown, R.J. Passonneau, D.K. Elson, A. Nenkova, and J. Hirschberg. Do summaries help? a task-based evaluation of multi-document summarization. In *Proceedings of the 28th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil*, page 6. Citeseer, 2005.

[Mihalcea and Tarau, 2004] R. Mihalcea and P. Tarau. TextRank–bringing order into texts. In *Proceedings of EMNLP*, pages 404–411, 2004.

[Nemhauser and Wolsey, 1988] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[Nemhauser *et al.*, 1978] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functionsâĂŤI. *Mathematical Programming*, 14(1):265–294, 1978.

[Nenkova and Passonneau, 2004] Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of HLT-NAACL*, 2004.

[Nenkova and Vanderwende, 2005] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research, Redmond, Washington, 2005.

[Nenkova *et al.*, 2006] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of SIGIR*, 2006.

[Nenkova, 2008] A. Nenkova. Entity-driven rewrite for multidocument summarization. In *Proceedings of IJCNLP*, 2008.

[Page *et al.*, 1998] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[Palmer and Hearst, 1997] D.D. Palmer and M.A. Hearst. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.

[Papineni *et al.*, 2002] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[Petrov and Klein, 2007] Slav Petrov and Dan Klein. Learning and inference for hierarchically split PCFGs. In *AAAI 2007 (Nectar Track)*, 2007.

[Rahman and Ng, 2011] A. Rahman and V. Ng. Narrowing the Modeling Gap: A Cluster-Ranking Approach to Coreference Resolution. *Journal of Artificial Intelligence Research*, 40:469–521, 2011.

[Reynar and Ratnaparkhi, 1997] J.C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, 1997.

[Schapire, 1999] R.E. Schapire. A brief introduction to boosting. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 1401–1406, 1999.

[Schilder and Kondadadi, 2008] Frank Schilder and Ravikumar Kondadadi. Fastsum: Fast and accurate query-based multi-document summarization. In *Proceedings of ACL-08: HLT, Short Papers*, 2008.

[Shen *et al.*, 2007] D. Shen, J.T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *the proceedings of IJCAI*, volume 7, pages 2862–2867, 2007.

[Snow *et al.*, 2008] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast, but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, 2008.

[Steyvers and Griffiths, 2007] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, page 427, 2007.

[Turner and Charniak, 2005] J. Turner and E. Charniak. Supervised and Unsupervised Learning for Sentence Compression. *Ann Arbor*, 100, 2005.

[Yandell, 1997] B.S. Yandell. *Practical data analysis for designed experiments*. Chapman & Hall/CRC, 1997.

[Yih *et al.*, 2007] W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007.