A Survey on Automatic Text Summarization

Dipanjan Das André F.T. Martins

Language Technologies Institute
Carnegie Mellon University
{dipanjan, afm}@cs.cmu.edu

November 21, 2007

Abstract

The increasing availability of online information has necessitated intensive research in the area of automatic text summarization within the Natural Language Processing (NLP) community. Over the past half a century, the problem has been addressed from many different perspectives, in varying domains and using various paradigms. This survey intends to investigate some of the most relevant approaches both in the areas of single-document and multiple-document summarization, giving special emphasis to empirical methods and extractive techniques. Some promising approaches that concentrate on specific details of the summarization problem are also discussed. Special attention is devoted to automatic evaluation of summarization systems, as future research on summarization is strongly dependent on progress in this area.

1 Introduction

The subfield of summarization has been investigated by the NLP community for nearly the last half century. Radev et al. (2002) define a *summary* as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". This simple definition captures three important aspects that characterize research on automatic summarization:

- Summaries may be produced from a *single document* or *multiple documents*,
- Summaries should preserve important information,
- Summaries should be short.

Even if we agree unanimously on these points, it seems from the literature that any attempt to provide a more elaborate definition for the task would result in disagreement within the community. In fact, many approaches differ on the manner of their problem formulations. We start by introducing some common terms in the

summarization dialect: *extraction* is the procedure of identifying important sections of the text and producing them verbatim; *abstraction* aims to produce important material in a new way; *fusion* combines extracted parts coherently; and *compression* aims to throw out unimportant sections of the text (Radev et al., 2002).

Earliest instances of research on summarizing scientific documents proposed paradigms for extracting salient sentences from text using features like *word* and *phrase frequency* (Luhn, 1958), *position* in the text (Baxendale, 1958) and *key phrases* (Edmundson, 1969). Various work published since then has concentrated on other domains, mostly on newswire data. Many approaches addressed the problem by building systems depending of the type of the required summary. While *extractive summarization* is mainly concerned with what the summary *content* should be, usually relying solely on extraction of sentences, *abstractive summarization* puts strong emphasis on the *form*, aiming to produce a grammatical summary, which usually requires advanced language generation techniques. In a paradigm more tuned to information retrieval (IR), one can also consider *topic-driven summarization*, that assumes that the summary content depends on the preference of the user and can be assessed via a *query*, making the final summary focused on a particular topic.

A crucial issue that will certainly drive future research on summarization is *evaluation*. During the last fifteen years, many system evaluation competitions like TREC,[1] DUC[2] and MUC[3] have created sets of training material and have established baselines for performance levels. However, a universal strategy to evaluate summarization systems is still absent.

In this survey, we primarily aim to investigate how empirical methods have been used to build summarization systems. The rest of the paper is organized as follows: Section 2 describes single-document summarization, focusing on extractive techniques. Section 3 progresses to discuss the area of multi-document summarization, where a few abstractive approaches that pioneered the field are also considered. Section 4 briefly discusses some unconventional approaches that we believe can be useful in the future of summarization research. Section 5 elaborates a few evaluation techniques and describes some of the standards for evaluating summaries automatically. Finally, Section 6 concludes the survey.

2 Single-Document Summarization

Usually, the flow of information in a given document is not uniform, which means that some parts are more important than others. The major challenge in summarization lies in distinguishing the more informative parts of a document from the less ones. Though there have been instances of research describing the automatic creation of *abstracts*, most work presented in the literature relies on verbatim *extraction* of sentences to address the problem of single-document summarization. In

[1] See `http://trec.nist.gov/`.

[2] See `http://duc.nist.gov/`.

[3] See `http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/muc_7_toc.html`.

this section, we describe some eminent extractive techniques. First, we look at early work from the 1950s and 60s that kicked off research on summarization. Second, we concentrate on approaches involving machine learning techniques published in the 1990s to today. Finally, we briefly describe some techniques that use a more complex natural language analysis to tackle the problem.

2.1 Early Work

Most early work on single-document summarization focused on *technical documents*. Perhaps the most cited paper on summarization is that of (Luhn, 1958), that describes research done at IBM in the 1950s. In his work, Luhn proposed that the *frequency* of a particular word in an article provides an useful measure of its significance. There are several key ideas put forward in this paper that have assumed importance in later work on summarization. As a first step, words were stemmed to their root forms, and stop words were deleted. Luhn then compiled a list of *content words* sorted by decreasing frequency, the index providing a significance measure of the word. On a sentence level, a *significance factor* was derived that reflects the number of occurrences of significant words within a sentence, and the linear distance between them due to the intervention of non-significant words. All sentences are ranked in order of their significance factor, and the top ranking sentences are finally selected to form the auto-abstract.

Related work (Baxendale, 1958), also done at IBM and published in the same journal, provides early insight on a particular feature helpful in finding salient parts of documents: the *sentence position*. Towards this goal, the author examined 200 paragraphs to find that in 85% of the paragraphs the topic sentence came as the first one and in 7% of the time it was the last sentence. Thus, a naive but fairly accurate way to select a topic sentence would be to choose one of these two. This positional feature has since been used in many complex machine learning based systems.

Edmundson (1969) describes a system that produces document extracts. His primary contribution was the development of a typical structure for an extractive summarization experiment. At first, the author developed a protocol for creating manual extracts, that was applied in a set of 400 technical documents. The two features of word frequency and positional importance were incorporated from the previous two works. Two other features were used: the presence of *cue words* (presence of words like *significant*, or *hardly*), and the *skeleton* of the document (whether the sentence is a title or heading). Weights were attached to each of these features manually to score each sentence. During evaluation, it was found that about 44% of the auto-extracts matched the manual extracts.

2.2 Machine Learning Methods

In the 1990s, with the advent of machine learning techniques in NLP, a series of seminal publications appeared that employed statistical techniques to produce document extracts. While initially most systems assumed feature independence and relied on naive-Bayes methods, others have focused on the choice of appropriate features and

on learning algorithms that make no independence assumptions. Other significant approaches involved hidden Markov models and log-linear models to improve extractive summarization. A very recent paper, in contrast, used neural networks and third party features (like common words in search engine queries) to improve purely extractive single document summarization. We next describe all these approaches in more detail.

2.2.1 Naive-Bayes Methods

Kupiec et al. (1995) describe a method derived from Edmundson (1969) that is able to learn from data. The classification function categorizes each sentence as worthy of extraction or not, using a *naive-Bayes classifier*. Let s be a particular sentence, $\mathcal{S}$ the set of sentences that make up the summary, and $F_1, \ldots, F_k$ the features. Assuming independence of the features:

$$P(s \in \mathcal{S} \mid F_1, F_2, ..F_k) = \frac{\prod_{i=1}^{k} P(F_i \mid s \in \mathcal{S}) \cdot P(s \in \mathcal{S})}{\prod_{i=1}^{k} P(F_i)} \tag{1}$$

The features were compliant to (Edmundson, 1969), but additionally included the *sentence length* and the *presence of uppercase words*. Each sentence was given a score according to (1), and only the n top sentences were extracted. To evaluate the system, a corpus of technical documents with manual abstracts was used in the following way: for each sentence in the manual abstract, the authors manually analyzed its match with the actual document sentences and created a mapping (e.g. exact match with a sentence, matching a join of two sentences, not matchable, etc.). The auto-extracts were then evaluated against this mapping. Feature analysis revealed that a system using only the position and the cue features, along with the sentence length sentence feature, performed best.

Aone et al. (1999) also incorporated a naive-Bayes classifier, but with richer features. They describe a system called DimSum that made use of features like term frequency (tf) and inverse document frequency (idf) to derive *signature words*.[4] The idf was computed from a large corpus of the same domain as the concerned documents. Statistically derived two-noun word collocations were used as units for counting, along with single words. A named-entity tagger was used and each entity was considered as a single token. They also employed some shallow discourse analysis like reference to same entities in the text, maintaining cohesion. The references were resolved at a very shallow level by linking name aliases within a document like "U.S." to "United States", or "IBM" for "International Business Machines". Synonyms and morphological variants were also merged while considering lexical terms, the former being identified by using Wordnet (Miller, 1995). The corpora used in the experiments were from newswire, some of which belonged to the TREC evaluations.

[4]Words that indicate key concepts in a document.

2.2.2 Rich Features and Decision Trees

Lin and Hovy (1997) studied the importance of a single feature, *sentence position*. Just weighing a sentence by its position in text, which the authors term as the "position method", arises from the idea that texts generally follow a predictable discourse structure, and that the sentences of greater topic centrality tend to occur in certain specifiable locations (e.g. title, abstracts, etc). However, since the discourse structure significantly varies over domains, the position method cannot be defined as naively as in (Baxendale, 1958). The paper makes an important contribution by investigating techniques of tailoring the position method towards optimality over a genre and how it can be evaluated for effectiveness. A newswire corpus was used, the collection of Ziff-Davis texts produced from the TIPSTER[5] program; it consists of text about computer and related hardware, accompanied by a set of key topic words and a small abstract of six sentences. For each document in the corpus, the authors measured the yield of each sentence position against the topic keywords. They then ranked the sentence positions by their average yield to produce the *Optimal Position Policy* (OPP) for topic positions for the genre.

Two kinds of evaluation were performed. Previously unseen text was used for testing whether the same procedure would work in a different domain. The first evaluation showed contours exactly like the training documents. In the second evaluation, word overlap of manual abstracts with the extracted sentences was measured. Windows in abstracts were compared with windows on the selected sentences and corresponding precision and recall values were measured. A high degree of coverage indicated the effectiveness of the position method.

In later work, Lin (1999) broke away from the assumption that features are independent of each other and tried to model the problem of sentence extraction using *decision trees*, instead of a naive-Bayes classifier. He examined a lot of features and their effect on sentence extraction. The data used in this work is a publicly available collection of texts, classified into various topics, provided by the TIPSTER-SUMMAC[6] evaluations, targeted towards information retrieval systems. The dataset contains essential text fragments (phrases, clauses, and sentences) which must be included in summaries to answer some TREC topics. These fragments were each evaluated by a human judge. The experiments described in the paper are with the SUMMARIST system developed at the University of Southern California. The system extracted sentences from the documents and those were matched against human extracts, like most early work on extractive summarization.

Some novel features were the *query signature* (normalized score given to sentences depending on number of query words that they contain), *IR signature* (the m most salient words in the corpus, similar to the signature words of (Aone et al., 1999)), *numerical data* (boolean value 1 given to sentences that contained a number in them), *proper name* (boolean value 1 given to sentences that contained a proper name in them), *pronoun or adjective* (boolean value 1 given to sentences

[5]See `http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/`.

[6]See `http://www-nlpir.nist.gov/related_projects/tipster_summac/index.html`.

that contained a pronoun or adjective in them), *weekday or month* (similar as previous feature) and *quotation* (similar as previous feature). It is worth noting that some features like the *query signature* are question-oriented because of the setting of the evaluation, unlike a generalized summarization framework.

The author experimented with various baselines, like using only the positional feature, or using a simple combination of all features by adding their values. When evaluated by matching machine extracted and human extracted sentences, the decision tree classifier was clearly the winner for the whole dataset, but for three topics, a naive combination of features beat it. Lin conjectured that this happened because some of the features were independent of each other. Feature analysis suggested that the IR signature was a valuable feature, corroborating the early findings of Luhn (1958).

2.2.3 Hidden Markov Models

In contrast with previous approaches, that were mostly feature-based and non-sequential, Conroy and O'leary (2001) modeled the problem of extracting a sentence from a document using a *hidden Markov model* (HMM). The basic motivation for using a sequential model is to account for *local dependencies* between sentences. Only three features were used: *position* of the sentence in the document (built into the state structure of the HMM), *number of terms* in the sentence, and *likeliness* of the sentence terms given the document terms.

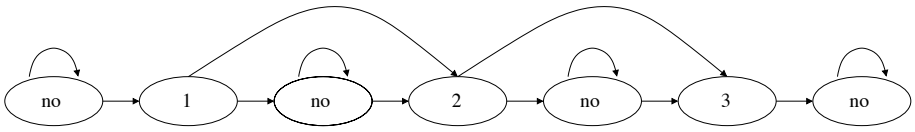

Figure 1: Markov model to extract to three summary sentences from a document (Conroy and O'leary, 2001).

The HMM was structured as follows: it contained $2s + 1$ states, alternating between s *summary states* and $s+1$ *nonsummary states*. The authors allowed "hesitation" only in nonsummary states and "skipping next state" only in summary states. Figure 1 shows an example HMM with 7 nodes, corresponding to $s = 3$. Using the TREC dataset as training corpus, the authors obtained the maximum-likelihood estimate for each transition probability, forming the transition matrix estimate $\hat{M}$, whose element (i, j) is the empirical probability of transitioning from state i to j. Associated with each state i was an output function, $b_i(O) = \Pr(O \mid \text{state } i)$ where O is an observed vector of features. They made a simplifying assumption that the features are multivariate normal. The output function for each state was thus estimated by using the training data to compute the maximum likelihood estimate of its mean and covariance matrix. They estimated $2s+1$ means, but assumed that all of the output functions shared a common covariance matrix. Evaluation was done

by comparing with human generated extracts.

2.2.4 Log-Linear Models

Osborne (2002) claims that existing approaches to summarization have always assumed feature independence. The author used log-linear models to obviate this assumption and showed empirically that the system produced better extracts than a naive-Bayes model, with a prior appended to both models. Let c be a label, s the item we are interested in labeling, f_i the i-th feature, and λ_i the corresponding feature weight. The conditional log-linear model used by Osborne (2002) can be stated as follows:

$$P(c \mid s) = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(c, s)\right), \tag{2}$$

where $Z(s) = \sum_c \exp\left(\sum_i \lambda_i f_i(c, s)\right)$. In this domain, there are only two possible labels: either the sentence is to be extracted or it is not. The weights were trained by conjugate gradient descent. The authors added a non-uniform prior to the model, claiming that a log-linear model tends to reject too many sentences for inclusion in a summary. The same prior was also added to a naive-Bayes model for comparison. The classification took place as follows:

$$\text{label}(s) = \arg\max_{c \in C} P(c) \cdot P(s, c) = \arg\max_{c \in C} \left(\log P(c) + \sum_i \lambda_i f_i(c, s)\right). \tag{3}$$

The authors optimized the prior using the $f2$ score of the classifier as an objective function on a part of the dataset (in the technical domain). The summaries were evaluated using the standard $f2$ score where $f2 = \frac{2pr}{p+r}$, where the precision and recall measures were measured against human generated extracts. The features included *word pairs* (pairs of words with all words truncated to ten characters), *sentence length*, *sentence position*, and naive discourse features like *inside introduction* or *inside conclusion*. With respect to $f2$ score, the log-linear model outperformed the naive-Bayes classifier with the prior, exhibiting the former's effectiveness.

2.2.5 Neural Networks and Third Party Features

In 2001-02, DUC issued a task of creating a 100-word summary of a single news article. However, the best performing systems in the evaluations could not outperform the baseline with statistical significance. This extremely strong baseline has been analyzed by Nenkova (2005) and corresponds to the selection of the first n sentences of a newswire article. This surprising result has been attributed to the journalistic convention of putting the most important part of an article in the initial paragraphs. After 2002, the task of single-document summarization for newswire was dropped from DUC. Svore et al. (2007) propose an algorithm based on neural nets and the use of third party datasets to tackle the problem of extractive summarization, outperforming the baseline with statistical significance.

The authors used a dataset containing 1365 documents gathered from CNN.com, each consisting of the title, timestamp, three or four human generated story highlights and the article text. They considered the task of creating three machine highlights. The human generated highlights were *not* verbatim extractions from the article itself. The authors evaluated their system using two metrics: the first one concatenated the three highlights produced by the system, concatenated the three human generated highlights, and compared these two *blocks*; the second metric considered the ordering and compared the sentences on an individual level.

Svore et al. (2007) trained a model from the labels and the features for each sentence of an article, that could infer the proper ranking of sentences in a test document. The ranking was accomplished using RankNet (Burges et al., 2005), a pair-based neural network algorithm designed to rank a set of inputs that uses the gradient descent method for training. For the training set, they used ROUGE-1 (Lin, 2004) to score the similarity of a human written highlight and a sentence in the document. These similarity scores were used as soft labels during training, contrasting with other approaches where sentences are "hard-labeled", as selected or not.

Some of the used features based on position or n-grams frequencies have been observed in previous work. However, the novelty of the framework lay in the use of features that derived information from query logs from Microsoft's news search engine[7] and Wikipedia[8] entries. The authors conjecture that if a document sentence contained keywords used in the news search engine, or entities found in Wikipedia articles, then there is a greater chance of having that sentence in the highlight. The extracts were evaluated using ROUGE-1 and ROUGE-2, and showed statistically significant improvements over the baseline of selecting the first three sentences in a document.

2.3 Deep Natural Language Analysis Methods

In this subsection, we describe a set of papers that detail approaches towards single-document summarization involving complex natural language analysis techniques. None of these papers solve the problem using machine learning, but rather use a set of heuristics to create document extracts. Most of these techniques try to model the text's discourse structure.

Barzilay and Elhadad (1997) describe a work that used considerable amount of linguistic analysis for performing the task of summarization. For a better understanding of their method, we need to define a *lexical chain*: it is a sequence of related words in a text, spanning short (adjacent words or sentences) or long distances (entire text). The authors' method progressed with the following steps: segmentation of the text, identification of lexical chains, and using strong lexical chains to identify the sentences worthy of extraction. They tried to reach a middle ground between (McKeown and Radev, 1995) and (Luhn, 1958) where the former relied on deep

[7] See `http://search.live.com/news`.

[8] See `http://en.wikipedia.org`.

semantic structure of the text, while the latter relied on word statistics of the documents. The authors describe the notion of *cohesion* in text as a means of sticking together different parts of the text. Lexical cohesion is a notable example where semantically related words are used. For example, let us take a look at the following sentence.[9]

$$John\ bought\ a\ \text{Jag}.\ He\ loves\ the\ \text{car}. \tag{4}$$

Here, the word *car* refers to the word *Jag* in the previous sentence, and exemplifies lexical cohesion. The phenomenon of cohesion occurs not only at the word level, but at word sequences too, resulting in lexical chains, which the authors used as a source representation for summarization. Semantically related words and word sequences were identified in the document, and several chains were extracted, that form a representation of the document. To find out lexical chains, the authors used Wordnet (Miller, 1995), applying three generic steps:

1. Selecting a set of candidate words.
2. For each candidate word, finding an appropriate chain relying on a relatedness criterion among members of the chains,
3. If it is found, inserting the word in the chain and updating it accordingly.

The relatedness was measured in terms of Wordnet distance. Simple nouns and noun compounds were used as starting point to find the set of candidates. In the final steps, strong lexical chains were used to create the summaries. The chains were scored by their length and homogeneity. Then the authors used a few heuristics to select the significant sentences.

In another paper, Ono et al. (1994) put forward a computational model of discourse for Japanese expository writings, where they elaborate a practical procedure for extracting the discourse *rhetorical structure*, a binary tree representing relations between chunks of sentences (rhetorical structure trees are used more intensively in (Marcu, 1998a), as we will see below). This structure was extracted using a series of NLP steps: sentence analysis, rhetorical relation extraction, segmentation, candidate generation and preference judgement. Evaluation was based on the relative importance of rhetorical relations. In the following step, the nodes of the rhetorical structure tree were pruned to reduce the sentence, keeping its important parts. Same was done for paragraphs to finally produce the summary. Evaluation was done with respect to sentence coverage and 30 editorial articles of a Japanese newspaper were used as the dataset. The articles had corresponding sets of key sentences and most important key sentences judged by human subjects. The key sentence coverage was about 51% and the most important key sentence coverage was 74%, indicating encouraging results.

Marcu (1998a) describes a unique approach towards summarization that, unlike most other previous work, does not assume that the sentences in a document form a flat sequence. This paper used discourse based heuristics with the traditional

[9]Example from `http://www.cs.ucd.ie/staff/jcarthy/home/Lex.html`.

features that have been used in the summarization literature. The discourse theory used in this paper is the Rhetorical Structure Theory (RST) that holds between two non-overlapping pieces of text spans: the *nucleus* and the *satellite*. The author mentions that the distinction between nuclei and satellites comes from the empirical observation that the nucleus expresses what is more essential to the writer's purpose than the satellite; and that the nucleus of a rhetorical relation is comprehensible independent of the satellite, but not vice versa. Marcu (1998b) describes the details of a rhetorical parser producing a discourse tree. Figure 2 shows an example discourse tree for a text example detailed in the paper. Once such a dis-

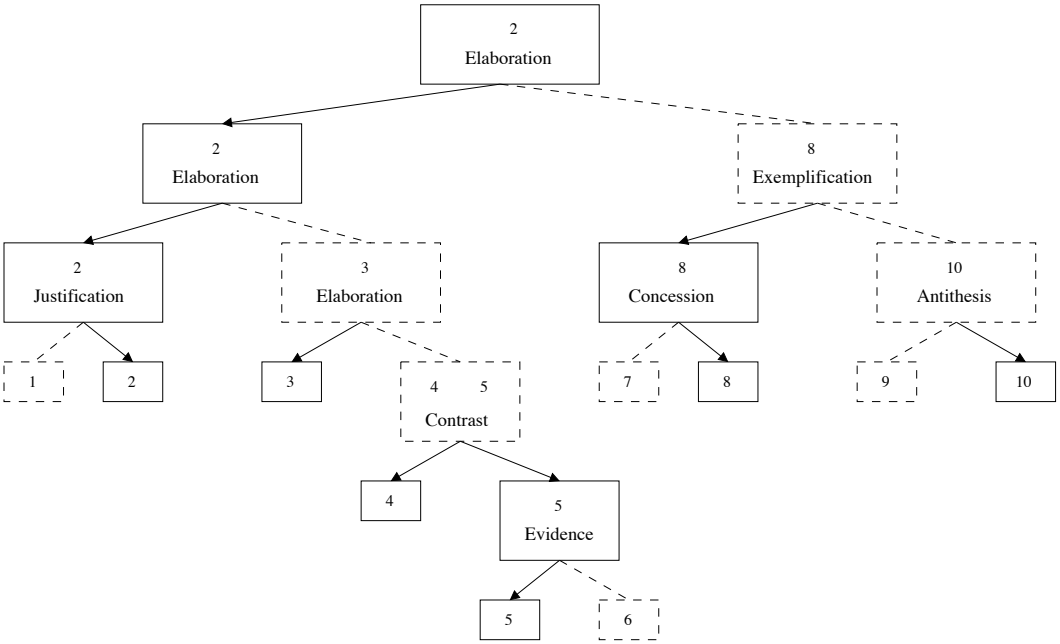

Figure 2: Example of a discourse tree from Marcu (1998a). The numbers in the nodes denote sentence numbers from the text example. The text below the number in selected nodes are rhetorical relations. The dotted nodes are SATELLITES and the normals ones are the NUCLEI.

course structure is created, a partial ordering of important units can be developed from the tree. Each equivalence class in the partial ordering is derived from the new sentences at a particular level of the discourse tree. In Figure 2, we observe that sentence 2 is at the root, followed by sentence 8 in the second level. In the third level, sentence 3 and 10 are observed, and so forth. The equivalence classes are $2 > 8 > 3, 10 > 1, 4, 5, 7, 9 > 6$.

If it is specified that the summary should contain the top $k\%$ of the text, the first $k\%$ of the units in the partial ordering can be selected to produce the summary. The author talks about a summarization system based just on this method in (Marcu, 1998b) and in one of his earlier papers. In this paper, he merged the discourse based heuristics with traditional heuristics. The metrics used were *clustering based*

metric (each node in the discourse tree was assigned a cluster score; for leaves the score was 0, for the internal nodes it was given by the similarity of the immediate children; discourse tree A was chosen to be better than B if its clustering score was higher), *marker based metric* (a discourse structure A was chosen to be better than a discourse structure B if A used more rhetorical relations than B), *rhetorical clustering based technique* (measured the similarity between salient units of two text spans), *shape based metric* (preferred a discourse tree A over B if A was more skewed towards the right than B), *title based metric*, *position based metric*, *connectedness based metric* (cosine similarity of an unit to all other text units, a discourse structure A was chosen to be better than B if its connectedness measure was more than B).

A weighted linear combination of all these scores gave the score of a discourse structure. To find the best combination of heuristics, the author computed the weights that maximized the F-score on the training dataset, which was constituted by newswire articles. To do this, he used a GSAT-like algorithm (Selman et al., 1992) that performed a greedy search in a seven dimensional space of the metrics. For a part of his corpus (the TREC dataset), a best F-score of 75.42% was achieved for the 10% summaries which was 3.5% higher than a baseline lead based algorithm, which was very encouraging.

3 Multi-Document Summarization

Extraction of a single summary from multiple documents has gained interest since mid 1990s, most applications being in the domain of news articles. Several Web-based news clustering systems were inspired by research on multi-document summarization, for example *Google News*,[10] *Columbia NewsBlaster*,[11] or *News In Essence*.[12] This departs from single-document summarization since the problem involves multiple sources of information that overlap and supplement each other, being contradictory at occasions. So the key tasks are not only identifying and coping with redundancy across documents, but also recognizing novelty and ensuring that the final summary is both coherent and complete.

The field seems to have been pioneered by the NLP group at Columbia University (McKeown and Radev, 1995), where a summarization system called SUMMONS[13] was developed by extending already existing technology for template-driven message understanding systems. Although in that early stage multi-document summarization was mainly seen as a task requiring substantial capabilities of both language interpretation and generation, it later gained autonomy, as people coming from different communities added new perspectives to the problem. Extractive techniques have been applied, making use of similarity measures between pairs of sentences. Approaches vary on how these similarities are used: some identify common themes through clustering and then select one sentence to represent each cluster (McKeown

[10] See `http://news.google.com`.
[11] See `http://newsblaster.cs.columbia.edu`.
[12] See `http://NewsInEssence.com`.
[13] SUMMarizing Online NewS articles.

et al., 1999; Radev et al., 2000), others generate a composite sentence from each cluster (Barzilay et al., 1999), while some approaches work dynamically by including each candidate passage only if it is considered novel with respect to the previous included passages, via *maximal marginal relevance* (Carbonell and Goldstein, 1998). Some recent work extends multi-document summarization to multilingual environments (Evans, 2005).

The way the problem is posed has also varied over time. While in some publications it is claimed that extractive techniques would not be effective for multi-document summarization (McKeown and Radev, 1995; McKeown et al., 1999), some years later that claim was overturned, as extractive systems like MEAD[14] (Radev et al., 2000) achieved good performance in large scale summarization of news articles. This can be explained by the fact that summarization systems often distinguish among themselves about what their goal actually is. While some systems, like SUMMONS, are designed to work in strict domains, aiming to build a sort of *briefing* that highlights differences and updates accross different news reports, putting much emphasis on *how* information is presented to the user, others, like MEAD, are large scale systems that intend to work in general domains, being more concerned with information *content* rather than *form*. Consequently, systems of the former kind require a strong effort on language generation to produce a grammatical and coherent summary, while latter systems are probably more close to the information retrieval paradigm. Abstractive systems like SUMMONS are difficult to replicate, as they heavily rely on the adaptation of internal tools to perform information extraction and language generation. On the other hand, extractive systems are generally easy to implement from scratch, and this makes them appealing when sophisticated NLP tools are not available.

3.1 Abstraction and Information Fusion

As far as we know, SUMMONS (McKeown and Radev, 1995; Radev and McKeown, 1998) is the first historical example of a multi-document summarization system. It tackles single events about a narrow domain (news articles about terrorism) and produces a *briefing* merging relevant information about each event and how reports by different news agencies have evolved over time. The whole thread of reports is then presented, as illustrated in the following example of a "good" summary:

> "In the afternoon of February 26, 1993, Reuters reported that a suspect bomb killed at least five people in the World Trade Center. However, Associated Press announced that exactly five people were killed in the blast. Finally, Associated Press announced that Arab terrorists were possibly responsible for the terrorist act."

Rather than working with raw text, SUMMONS reads a database previously built by a template-based message understanding system. A full multi-document

[14] Available for download at `http://www.summarization.com/mead/`.

summarizer is built by concatenating the two systems, first processing full text as input and filling *template slots*, and then synthesizing a summary from the extracted information. The architecture of SUMMONS consists of two major components: a *content planner* that selects the information to include in the summary through combination of the input templates, and a *linguistic generator* that selects the right words to express the information in grammatical and coherent text. The latter component was devised by adapting existing language generation tools, namely the FUF/SURGE system[15]. Content planning, on the other hand, is made through *summary operators*, a set of heuristic rules that perform operations like "change of perspective", "contradiction", "refinement", etc. Some of these operations require resolving *conflicts*, i.e., contradictory information among different sources or time instants; others complete pieces of information that are included in some articles and not in others, combining them into a single template. At the end, the linguistic generator gathers all the combined information and uses connective phrases to synthesize a summary.

While this framework seems promising when the domain is narrow enough so that the templates can be designed by hand, a generalization for broader domains would be problematic. This was improved later by McKeown et al. (1999) and Barzilay et al. (1999), where the input is now a set of related documents in raw text, like those retrieved by a standard search engine in response to a query. The system starts by identifying *themes*, i.e., sets of similar text units (usually paragraphs). This is formulated as a *clustering* problem. To compute a similarity measure between text units, these are mapped to vectors of features, that include single words weighted by their TF-IDF scores, noun phrases, proper nouns, *synsets* from the Wordnet database and a database of semantic classes of verbs. For each pair of paragraphs, a vector is computed that represents matches on the different features. Decision rules that were learned from data are then used to classify each pair of text units either as *similar* or *dissimilar*; this in turn feeds a subsequent algorithm that places the most related paragraphs in the same *theme*.

Once themes are identified, the system enters its second stage: *information fusion*. The goal is to decide which sentences of a theme should be included in the summary. Rather than just picking a sentence that is a group representative, the authors propose an algorithm which compares and intersects predicate argument structures of the phrases within each theme to determine which are repeated often enough to be included in the summary. This is done as follows: first, sentences are parsed through Collins' statistical parser (Collins, 1999) and converted into *dependency trees*, which allows capturing the predicate-argument structure and identify functional roles. Determiners and auxiliaries are dropped; Fig. 3 shows a sentence representation.

The comparison algorithm then traverses these dependency trees recursively, adding identical nodes to the output tree. Once full phrases (a verb with at least two constituents) are found, they are marked to be included in the summary. If two

[15]FUF, SURGE, and other tools developed by the Columbia NLP group are available at `http://www1.cs.columbia.edu/nlp/tools.cgi`.

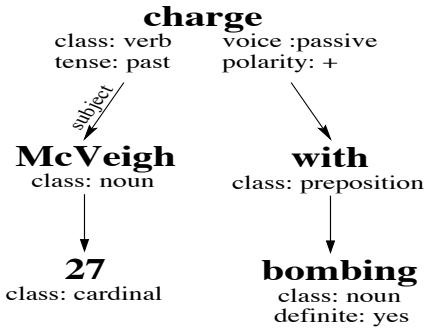

Figure 3: Dependency tree representing the sentence "McVeigh, 27, was charged with the bombing" (extracted from (McKeown et al., 1999)).

phrases, rooted at some node, are not identical but yet similar, the hypothesis that they are *paraphrases* of each other is considered; to take this into account, corpus-driven paraphrasing rules are written to allow paraphrase intersection.[16] Once the summary content (represented as predicate-argument structures) is decided, a grammatical text is generated by translating those structures into the arguments expected by the FUF/SURGE language generation system.

3.2 Topic-driven Summarization and MMR

Carbonell and Goldstein (1998) made a major contribution to topic-driven summarization by introducing the *maximal marginal relevance* (MMR) measure. The idea is to combine *query relevance* with *information novelty*; it may be applicable in several tasks ranging from text retrieval to topic-driven summarization. MMR simultaneously rewards relevant sentences and penalizes redundant ones by considering a linear combination of two similarity measures.

Let Q be a query or user profile and R a ranked list of documents retrieved by a search engine. Consider an incremental procedure that selects documents, one at a time, and adds them to a set S. So let S be the set of already selected documents in a particular step, and $R \setminus S$ the set of yet unselected documents in R. For each candidate document $D_i \in R \setminus S$, its *marginal relevance* $\mathrm{MR}(D_i)$ is computed as:

$$\mathrm{MR}(D_i) := \lambda \mathrm{Sim}_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} \mathrm{Sim}_2(D_i, D_j) \tag{5}$$

where λ is a parameter lying in $[0, 1]$ that controls the relative importance given to *relevance* versus *redundancy*. Sim_1 and Sim_2 are two similarity measures; in the

[16] A full description of the kind of paraphrasing rules used can be found in (Barzilay et al., 1999). Examples are: ordering of sentence components, main clause vs. relative clause, realization in different syntactic categories (e.g. classifier vs. apposition), change in grammatical features (active/passive, time, number, etc.), head omission, transformation from one POS to another, using semantically related words (e.g. synonyms), etc.

experiments both were set to the standard cosine similarity traditionally used in the vector space model, $\mathrm{Sim}_1(x, y) = \mathrm{Sim}_2(x, y) = \frac{\langle x,y \rangle}{\|x\| \cdot \|y\|}$. The document achieving the highest marginal relevance, $D_{\mathrm{MMR}} = \arg\max_{D_i \in R \setminus S} \mathrm{MR}(D_i)$, is then selected, i.e., added to S, and the procedure continues until a maximum number of documents are selected or a minimum relevance threshold is attained. Carbonell and Goldstein (1998) found experimentally that choosing dynamically the value of λ turns out to be more effective than keeping it fixed, namely starting with small values ($\lambda \approx 0.3$) to give more emphasis to novelty, and then increasing it ($\lambda \approx 0.7$) to focus on the most relevant documents. To perform summarization, documents can be first segmented into sentences or paragraphs, and after a query is submitted, the MMR algorithm can be applied followed by a selection of the top ranking passages, reordering them as they appeared in the original documents, and presenting the result as the summary.

One of the attractive points in using MMR for summarization is its topic-oriented feature, through its dependency on the query Q, which makes it particularly appealing to generate summaries according to a *user profile*: as the authors claim, "a different user with different information needs may require a totally different summary of the same document." This assertion was not being taken into account by previous multi-document summarization systems.

3.3 Graph Spreading Activation

Mani and Bloedorn (1997) describe an information extraction framework for summarization, a graph-based method to find similarities and dissimilarities in pairs of documents. Albeit no textual summary is generated, the summary *content* is represented via entities (*concepts*) and *relations* that are displayed respectively as nodes and edges of a graph. Rather than extracting sentences, they detect salient *regions* of the graph via a *spreading activation* technique.[17]

This approach shares with the method described in Section 3.2 the property of being topic-driven; there is an additional input that stands for the *topic* with respect to which the summary is to be generated. The topic is represented through a set of *entry nodes* in the graph. A document is represented as a graph as follows: each node represents the *occurrence* of a single word (i.e., one word together with its position in the text). Each node can have several kinds of links: *adjacency links* (ADJ) to adjacent words in the text, SAME links to other occurrences of the same word, and ALPHA links encoding semantic relationships captured through Wordnet and NetOwl[18]. Besides these, PHRASE links tie together sequences of adjacent nodes which belong to the same phrase, and NAME and COREF links stand for co-referential name occurrences; Fig. 4 shows some of these links.

Once the graph is built, *topic nodes* are identified by stem comparison and become the *entry nodes*. A search for semantically related text is then propagated from these to the other nodes of the graph, in a process called *spreading activation*. Salient

[17] The name "spreading activation" is borrowed from a method used in information retrieval (Salton and Buckley, 1988) to expand the search vocabulary.

[18] See `http://www.netowl.com`.

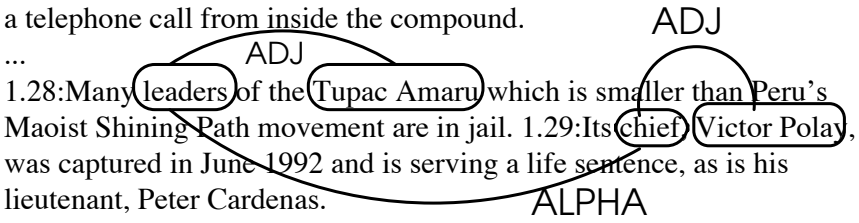

Figure 4: Examples of nodes and links in the graph for a particular sentence (detail extracted from from a figure in (Mani and Bloedorn, 1997)).

words and phrases are initialized according to their TF-IDF score. The weight of neighboring nodes depends on the node link traveled and is an exponentially decaying function of the distance of the traversed path. Traveling within a sentence is made cheaper than across sentence boundaries, which in turn is cheaper than across paragraph boundaries. Given a pair of document graphs, *common nodes* are identified either by sharing the same stem or by being synonyms. Analogously, *difference nodes* are those that are not common. For each sentence in both documents, two scores are computed: one score that reflects the presence of common nodes, which is computed as the average weight of these nodes; and another score that computes instead the average weights of difference nodes. Both scores are computed *after spreading activation*. In the end, the sentences that have higher common and different scores are highlighted, the user being able to specify the maximal number of common and different sentences to control the output. In the future, the authors expect to use these structure to actually compose abstractive summaries, rather than just highlighting pieces of text.

3.4 Centroid-based Summarization

Although clustering techniques were already being employed by McKeown et al. (1999) and Barzilay et al. (1999) for identification of themes, Radev et al. (2000) pioneered the use of cluster *centroids* to play a central role in summarization. A full description of the centroid-based approach that underlies the MEAD system can be found in (Radev et al., 2004); here we sketch briefly the main points. Perhaps the most appealing feature is the fact that it does not make use of any language generation module, unlike most previous systems. All documents are modeled as bags-of-words. The system is also easily scalable and domain-independent.

The first stage consists of topic detection, whose goal is to group together news articles that describe the same event. To accomplish this task, an agglomerative clustering algorithm is used that operates over the TF-IDF vector representations of the documents, successively adding documents to clusters and recomputing the

centroids according to

$$\mathbf{c}_j = \frac{\sum_{\mathbf{d} \in C_j} \tilde{\mathbf{d}}}{|C_j|} \tag{6}$$

where $\mathbf{c}_j$ is the centroid of the j-th cluster, C_j is the set of documents that belong to that cluster, its cardinality being $|C_j|$, and $\tilde{\mathbf{d}}$ is a "truncated version" of $\mathbf{d}$ that vanishes on those words whose TF-IDF scores are below a threshold. *Centroids* can thus be regarded as pseudo-documents that include those words whose TF-IDF scores are above a threshold in the documents that constitute the cluster. Each *event cluster* is a collection of (typically 2 to 10) news articles from multiple sources, chronologically ordered, describing an event as it develops over time.

The second stage uses the centroids to identify sentences in each cluster that are central to the topic of the entire cluster. In (Radev et al., 2000), two metrics are defined that resemble the two summands in the MMR (see Section 3.2): *cluster-based relative utility* (CBRU) and *cross-sentence informational subsumption* (CSIS). The first accounts for how relevant a particular sentence is to the general topic of the entire cluster; the second is a measure of redundancy among sentences. Unlike MMR, these metrics are not query-dependent. Given one cluster C of documents segmented into n sentences, and a compression rate R, a sequence of nR sentences are extracted in the same order as they appear in the original documents, which in turn are ordered chronologically. The selection of the sentences is made by approximating their CBRU and CSIS.[19] For each sentence s_i, three different features are used:

- Its *centroid value* (C_i), defined as the sum of the centroid values of all the words in the sentence,
- A *positional value* (P_i), that is used to make leading sentences more important. Let $C_{\max}$ be the centroid value of the highest ranked sentence in the document. Then $P_i = \frac{n-i+1}{n} C_{\max}$.
- The *first-sentence overlap* (F_i), defined as the inner product between the word occurrence vector of sentence i and that of the first sentence of the document.

The final score of each sentence is a combination of the three scores above minus a redundancy penalty (R_s) for each sentence that overlaps highly ranked sentences.

3.5 Multilingual Multi-document Summarization

Evans (2005) addresses the task of summarizing documents written in multiple languages; this had already been sketched by Hovy and Lin (1999). Multilingual summarization is still at an early stage, but this framework looks quite useful for newswire applications that need to combine information from foreign news agencies. Evans (2005) considered the scenario where there is a preferred language in which the summary is to be written, and multiple documents in the preferred and

[19] The two metrics are used directly for evaluation (see (Radev et al., 2004) for more details).

in foreign languages are available. In their experiments, the preferred language was English and the documents are news articles in English and Arabic. The rationale is to summarize the English articles without discarding the information contained in the Arabic documents. The IBM's statistical machine translation system is first applied to translate the Arabic documents to English. Then a search is made, for each translated text unit, to see whether there is a similar sentence or not in the English documents. If so, and if the sentence is found relevant enough to be included in the summary, the similar English sentence is included instead of the Arabic-to-English translation. This way, the final summary is more likely to be grammatical, since machine translation is known to be far from perfect. On the other hand, the result is also expected to have higher coverage than using just the English documents, since the information contained in the Arabic documents can help to decide about the relevance of each sentence. In order to measure similarity between sentences, a tool named *SimFinder*[20] was employed: this is a tool for clustering text based on similarity over a variety of lexical and syntactic features using a log-linear regression model.

4 Other Approaches to Summarization

This section describes briefly some unconventional approaches that, rather than aiming to build full summarization systems, investigate some details that underlie the summarization process, and that we conjecture to have a role to play in future research on this field.

4.1 Short Summaries

Witbrock and Mittal (1999) claim that extractive summarization is not very powerful in that the extracts are not concise enough when very short summaries are required. They present a system that generated headline style summaries. The corpus used in this work was newswire articles from Reuters and the Associated Press, publicly available at the LDC[21]. The system learned statistical models of the relationship between source text units and headline units. It attempted to model both the order and the likelihood of the appearance of tokens in the target documents. Both the models, one for content selection and the other for surface realization were used to co-constrain each other during the search in the summary generation task.

For content selection, the model learned a translation model between a document and its summary (Brown et al., 1993). This model in the simplest case can be thought as a mapping between a word in the document and the likelihood of some word appearing in the summary. To simplify the model, the authors assumed that the probability of a word appearing in a summary is independent of its structure. This mapping boils down to the fact that the probability of a particular summary

[20] See `http://www1.cs.columbia.edu/nlp/tools.cgi#SimFinder`.
[21] See `http://ldc.upenn.edu`.

candidate is the product of the probabilities of the summary content and that content being expressed using a particular structure.

The surface realization model used was a bigram model. Viterbi beam search was used to efficiently find a near-optimal summary. The Markov assumption was violated by using backtracking at every state to strongly discourage paths that repeated terms, since bigrams that start repeating often seem to pathologically overwhelm the search otherwise. To evaluate the system, the authors compared its output against the actual headlines for a set of input newswire stories. Since phrasing could not be compared, they compared the generated headlines against the actual headlines, as well as the top ranked summary sentence of the story. Since the system did not have a mechanism to determine the optimal length of a headline, six headlines for each story were generated, ranging in length from 4 to 10 words and they measured the term-overlap between each of the generated headlines and the test. For headline length 4, there was 0.89 overlap in the headline and there was 0.91 overlap amongst the top scored sentence, indicating useful results.

4.2 Sentence Compression

Knight and Marcu (2000) introduced a statistical approach to *sentence compression*. The authors believe that understanding the simpler task of compressing a sentence may be a fruitful first step to later tackle the problems of single and multi-document summarization.

Sentence compression is defined as follows: given a sequence of words $W = w_1 w_2 \dots w_n$ that constitute a sentence, find a subsequence $w_{i_1} w_{i_2} \dots w_{i_k}$, with $1 \leq i_1 < i_2 < \dots i_k \leq n$, that is a *compressed version* of W. Note that there are 2^n possibilities of output. Knight and Marcu (2000) considered two different approaches: one that is inspired by the *noisy-channel model*, and another one based on *decision trees*. Due to its simplicity and elegance, we describe the first approach here.

The noisy-channel model considers that one starts with a short summary s, drawn according to the source model $P(s)$, which is then subject to channel noise to become the full sentence t, in a process guided by the channel model $P(t|s)$. When the string t is observed, one wants to recover the original summary according to:

$$\hat{s} = \arg\max_s P(s|t) = \arg\max_s P(s)P(t|s). \tag{7}$$

This model has the advantage of decoupling the goals of producing a short text that looks grammatical (incorporated in the source model) and of preserving important information (which is done through the channel model). In (Knight and Marcu, 2000), the source and channel models are simple models inspired by *probabilistic context-free grammars* (PCFGs). The following probability mass functions are defined over *parse trees* rather than strings: $P_{\text{tree}}(s)$, the probability of a parse tree that generates s, and $P_{\text{expand\_tree}}(t|s)$, the probability that a small parse tree that generates s is *expanded* to a longer one that generates t.

The sentence t is first parsed by using Collins' parser (Collins, 1999). Then, rather than computing $P_{\text{tree}}(s)$ over all the 2^n hypotheses for s, which would be exponential in the sentence length, a *shaded-forest structure* is used: the parse tree of t is traversed and the grammar (learned from the Penn Treebank[22]) is used to check recursively which nodes may be removed from each production in order to achieve another valid production. This algorithm allows to compute efficiently $P_{\text{tree}}(s)$ and $P_{\text{expand\_tree}}(t|s)$ for all possible *grammatical* summaries s. Conceptually, the noisy channel model works the other way around: summaries are the original strings that are expanded via *expansion templates*. Expansion operations have the effect of decreasing the probability $P_{\text{expand\_tree}}(t|s)$. The probabilities $P_{\text{tree}}(s)$ and $P_{\text{expand\_tree}}(t|s)$ consist in the usual factorized expression for PCFGs times a bigram distribution over the leaves of the tree (i.e. the words). In the end, the log probability is (heuristically) divided by the length of the sentence s in order not to penalize excessively longer sentences (this is done commonly in speech recognition).

More recently, Daumé III and Marcu (2002) extended this approach to *document* compression by using *rhetorical structure theory* as in Marcu (1998a), where the entire document is represented as a tree, hence allowing not only to compress relevant sentences, but also to drop irrelevant ones. In this framework, Daumé III and Marcu (2004) employed kernel methods to decide for each node in the tree whether or not it should be kept.

4.3 Sequential document representation

We conclude this section by mentioning some recent work that concerns document representation, with applications in summarization. In the bag-of-words representation (Salton et al., 1975) each document is represented as a sparse vector in a very large Euclidean space, indexed by words in the vocabulary V. A well-known technique in information retrieval to capture word correlation is *latent semantic indexing* (LSI), that aims to find a linear subspace of dimension $k \leq |V|$ where documents may be approximately represented by their projections.

These classical approaches assume by convenience that Euclidean geometry is a proper model for text documents. As an alternative, Gous (1999) and Hall and Hofmann (2000) used the framework of *information geometry* (Amari and Nagaoka, 2001) to generalize LSI to the *multinomial manifold*, which can be identified with the *probability simplex*

$$\mathbb{P}^{n-1} = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1,\ x_i \geq 0 \text{ for } i = 1, \ldots, n \right\}. \tag{8}$$

Instead of finding a linear subspace, as in the Euclidean case, they learn a submanifold of $\mathbb{P}^{n-1}$. To illustrate this idea, Gous (1999) split a book (Machiavelli's *The Prince*) into several text blocks (its numbered pages), considered each page as a point in $\mathbb{P}^{|V|-1}$, and projected data into a 2-dimensional submanifold. The result is

[22] See `http://www.cis.upenn.edu/~treebank/`.

the representation of the book as a sequential path in $\mathbb{R}^2$, tracking the evolution of the subject matter of the book over the course of its pages (see Fig. 5). Inspired by

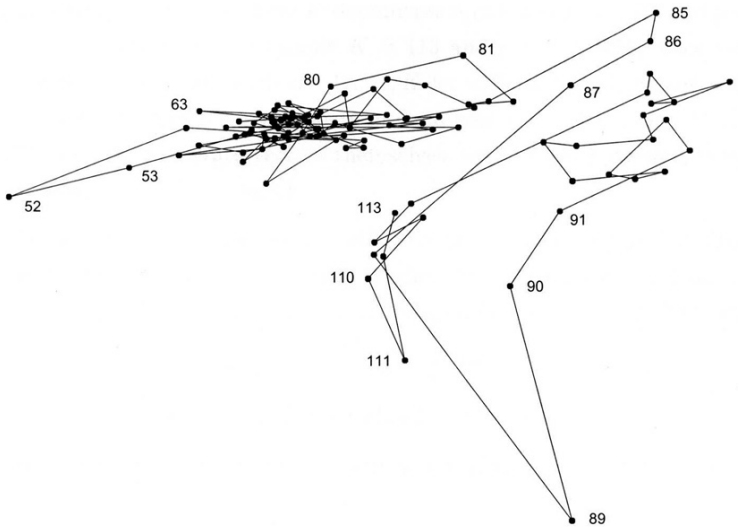

Figure 5: The 113 pages of *The Prince* projected onto a 2-dimensional space (extracted from (Gous, 1999)). The inflection around page 85 reflects a real change in the subject matter, where the book shifts from political theory to a more biographical discourse.

this framework, Lebanon et al. (2007) suggested representing a document as a *simplicial curve* (i.e. a curve in the probability simplex), yielding the *locally weighted bag-of-words* (lowbow) model. According to this representation, a length-normalized document is a *function* $x : [0, 1] \times V \to \mathbb{R}_+$ such that

$$\sum_{w_j \in V} x(t, w_j) = 1, \quad \text{for any } t \in [0, 1]. \tag{9}$$

We can regard the document as a continuous signal, and $x(t, w_j)$ as expressing the relevance of word w_j at instant t. This generalizes both the pure sequential representation and the (global) bag-of-words model. Let $y = (y_1, \ldots, y_n) \in V^n$ be a n-length document. The pure sequential representation of y arises by defining $x = x^{\text{seq}}$ with:

$$x^{\text{seq}}(t, w_j) = \begin{cases} 1, & \text{if } w_j = y_{\lceil tn \rceil} \\ 0, & \text{if } w_j \neq y_{\lceil tn \rceil}, \end{cases} \tag{10}$$

where $\lceil a \rceil$ denotes the smallest integer greater than a. The global bag-of-words representation of x corresponds to defining $x = x^{\text{bow}}$, where

$$x^{\text{bow}}(\mu, w_j) = \int_0^1 x^{\text{seq}}(t, w_j) dt, \quad \mu \in [0, 1], \ j = 1, \ldots, |V|. \tag{11}$$

In this case, the curve degenerates into a single point in the simplex, which is the maximum likelihood estimate of the multinomial parameters. An intermediate

representation arises by smoothing (10) via a function $f_{\mu,\sigma} : [0,1] \to \mathbb{R}_{++}$, where $\mu \in [0,1]$ and $\sigma \in \mathbb{R}_{++}$ are respectively a location and a scale parameter. An example of such a smoothing function is the truncated Gaussian defined in $[0,1]$ and normalized. This allows defining the lowbow representation at μ of the n-lenght document $(y_1, \ldots, y_n) \in V^n$ as the function $x : [0,1] \times V \to \mathbb{R}_+$ such that:

$$x(\mu, w_j) = \int_0^1 x^{\text{seq}}(t, w_j) f_{\mu,\sigma}(t) dt. \tag{12}$$

The scale of the smoothing function controls the amount of locality/globality in the document representation (see Fig. 6): when $\sigma \to \infty$ we recover the global bow representation (11); when $\sigma \to 0$, we approach the pure sequential representation (10).

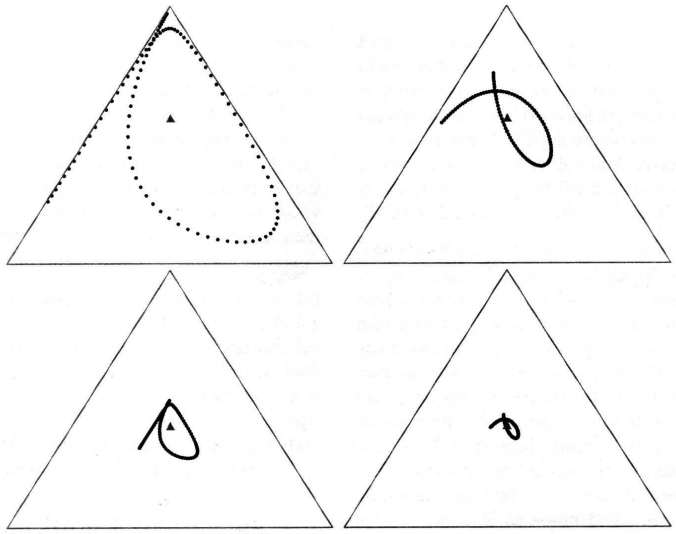

Figure 6: The lowbow representation of a document with $|V| = 3$, for several values of the scale parameter σ (extracted from (Lebanon, 2006)).

Representing a document as a simplicial curve allows us to characterize geometrically several properties of the document. For example, the tangent vector field along the curve describes sequential "topic trends" and their change; the curvature measures the amount of wigglyness or deviation from a geodesic path. This properties can be useful for tasks like text segmentation or summarization; for example plotting the velocity of the curve $||\dot{x}(\mu)||$ along time offers a visualization of the document where local maxima tend to correspond to topic boundaries (see (Lebanon et al., 2007) for more information).

5 Evaluation

Evaluating a summary is a difficult task because there does not exist an ideal summary for a given document or set of documents. From papers surveyed in the previous sections and elsewhere in literature, it has been found that agreement between human summarizers is quite low, both for evaluating and generating summaries. More than the form of the summary, it is difficult to evaluate the summary content. Another important problem in summary evaluation is the widespread use of disparate metrics. The absence of a standard human or automatic evaluation metric makes it very hard to compare different systems and establish a baseline. This problem is not present in other NLP problems, like parsing. Besides this, manual evaluation is too expensive: as stated by Lin (2004), large scale manual evaluation of summaries as in the DUC conferences would require over 3000 hours of human efforts. Hence, an evaluation metric having high correlation with human scores would obviate the process of manual evaluation. In this section, we would look at some important recent papers that have been able to create standards in the summarization community.

5.1 Human and Automatic Evaluation

Lin and Hovy (2002) describe and compare various human and automatic metrics to evaluate summaries. They focus on the evaluation procedure used in the Document Understanding Conference 2001 (DUC-2001), where the Summary Evaluation Environment (SEE) interface was used to support the human evaluation part. NIST assessors in DUC-2001 compared manually written *ideal summaries* with summaries generated automatically by summarization systems and baseline summaries. Each text was decomposed into a list of units (sentences) and displayed in separate windows in SEE. To measure the content of summaries, assessors stepped through each *model unit* (MU) from the ideal summaries and marked all *system units* (SU) sharing content with the current model unit, rating them with scores in the range $1-4$ to specify that the marked system units express *all* (4), *most* (3), *some* (2) or *hardly any* (1) of the content of the current model unit. Grammaticality, cohesion, and coherence were also rated similarly by the assessors. The *weighted recall* at threshold t (where t range from 1 to 4) is then defined as

$$\text{Recall}_t = \frac{\text{Number of MUs marked at or above } t}{\text{Number of MUs in the model summary}}. \tag{13}$$

An interesting study is presented that shows how unstable the human markings for overlapping units are. For multiple systems, the coverage scores assigned to the same units were different by human assessors 18% of the time for the single document task and 7.6% of the time for multi-document task. The authors also observe that inter-human agreement is quite low in creating extracts from documents ($\sim 40\%$ for single-documents and $\sim 29\%$ for multi-documents). To overcome the instability of human evaluations, they proposed using automatic metrics for summary evaluation.

Inspired by the machine translation evaluation metric BLEU (Papineni et al., 2001), they outline an *accumulative n-gram matching score* (which they call NAMS),

$$\text{NAMS} = \text{a}_1 \cdot \text{NAM}_1 + \text{a}_2 \cdot \text{NAM}_2 + \text{a}_3 \cdot \text{NAM}_3 + \text{a}_4 \cdot \text{NAM}_4, \tag{14}$$

where the NAM_n n-gram hit ratio is defined as:

$$\frac{\text{\# of matched } n\text{-grams between MU and S}}{\text{total \# of } n\text{-grams in MU}} \tag{15}$$

with S denoting here the *whole* system summary, and where only content words were used in forming the n-grams. Different configurations of a_i were tried; the best correlation with human judgement (using Spearman's rank order correlation coefficient) was achieved using a configuration giving 2/3 weight to bigram matches and 1/3 to unigrams matches with stemming done by the Porter stemmer.

5.2 ROUGE

Lin (2004) introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[23] that have become standards of automatic evaluation of summaries.

In what follows, let $R = \{r_1, \ldots, r_m\}$ be a set of *reference summaries*, and let s be a summary generated automatically by some system. Let $\Phi_n(d)$ be a binary vector representing the n-grams contained in a document d; the i-th component $\phi_n^i(d)$ is 1 if the i-th n-gram is contained in d and 0 otherwise. The metric ROUGE-N is an n-gram recall based statistic that can be computed as follows:

$$\text{ROUGE-N}(s) = \frac{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(s) \rangle}{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(r) \rangle}, \tag{16}$$

where $\langle ., . \rangle$ denotes the usual inner product of vectors. This measure is closely related to BLEU which is a precision related measure. Unlike other measures previously considered, ROUGE-N can be used for multiple reference summaries, which is quite useful in practical situations. An alternative is taking the most similar summary in the reference set,

$$\text{ROUGE-N}_{multi}(s) = \max_{r \in R} \frac{\langle \Phi_n(r), \Phi_n(s) \rangle}{\langle \Phi_n(r), \Phi_n(r) \rangle}. \tag{17}$$

Another metric in (Lin, 2004) applies the concept of *longest common subsequences*[24] (LCS). The rationale is: the longer the LCS between two summary sentences, the more similar they are. Let $r_1, \ldots, r_u$ be the reference sentences of the documents in R, and s a candidate summary (considered as a concatenation of sentences). The ROUGE-L is defined as an LCS based F-measure:

$$\text{ROUGE-L}(s) = \frac{(1 + \beta^2) R_{\text{LCS}} P_{\text{LCS}}}{R_{\text{LCS}} + \beta^2 P_{\text{LCS}}} \tag{18}$$

[23] See `http://openrouge.com/default.aspx`.

[24] A *subsequence* of a string $s = s_1 \ldots s_n$ is a string of the form $s_{i_1} \ldots s_{i_n}$ where $1 \leq i_1 < \ldots i_n \leq n$.

where $R_{\mathrm{LCS}}(s) = \frac{\sum_{i=1}^{u} LCS(r_i,s)}{\sum_{i=1}^{u} |r_i|}$, $P_{\mathrm{LCS}}(s) = \frac{\sum_{i=1}^{u} LCS(r_i,s)}{|s|}$, $|x|$ denotes the length of sentence x, $LCS(x,y)$ denotes the length of the LCS between sentences x and y, and β is a (usually large) parameter to balance precision and recall. Notice that the LCS function may be computed by a simple dynamic programming approach. The metric (18) is further refined by including weights that penalize subsequence matches that are not consecutive, yielding a new measure denoted ROUGE-W.

Yet another measure introduced by Lin (2004) is ROUGE-S, which can be seen as a gappy version of ROUGE-N for $n = 2$ and is aptly called *skip bigram*. Let $\Psi_2(d)$ be a binary vector indexed by ordered pairs of words; the i-th component $\psi_2^i(d)$ is 1 if the i-th pair is a subsequence of d and 0 otherwise. The metric ROUGE-S is computed as follows:

$$\text{ROUGE-S}(s) = \frac{(1+\beta^2) R_{\mathrm{S}} P_{\mathrm{S}}}{R_{\mathrm{S}} + \beta^2 P_{\mathrm{S}}} \tag{19}$$

where $R_{\mathrm{S}}(s) = \frac{\sum_{i=1}^{u} \langle \Psi_2(r_i), \Psi_2(s) \rangle}{\sum_{i=1}^{u} \langle \Psi_2(r_i), \Psi_2(r_i) \rangle}$ and $P_{\mathrm{S}}(s) = \frac{\sum_{i=1}^{u} \langle \Psi_2(r_i), \Psi_2(s) \rangle}{\langle \Psi_2(s), \Psi_2(s) \rangle}$.

The various versions of ROUGE were evaluated by computing the correlation coefficient between ROUGE scores and human judgement scores. ROUGE-2 performed the best among the ROUGE-N variants. ROUGE-L, ROUGE-W, and ROUGE-S all performed very well on the DUC-2001 and DUC-2002 datasets. However, correlation achieved with human judgement for multi-document summarization was not as high as single-document ones; improvement on this side of the paradigm is an open research topic.

5.3 Information-theoretic Evaluation of Summaries

A very recent approach (Lin et al., 2006) proposes to use an information-theoretic method to automatic evaluation of summaries. The central idea is to use a divergence measure between a pair of probability distributions, in this case the *Jensen-Shannon divergence*, where the first distribution is derived from an automatic summary and the second from a set of reference summaries. This approach has the advantage of suiting both the single-document and the multi-document summarization scenarios.

Let $D = \{d_1, \ldots, d_n\}$ be the set of documents to summarize (which is a singleton set in the case of single-document summarization). Assume that a distribution parameterized by θ_R generates reference summaries of the documents in D. The task of summarization can be seen as that of estimating θ_R. Analogously, assume that every summarization system is governed by some distribution parameterized by θ_A. Then, we may define a *good summarizer* as one for which θ_A is close to θ_R. One information-theoretic measure between distributions that is adequate for this is the KL divergence (Cover and Thomas, 1991),

$$KL(p^{\theta_A} || p^{\theta_R}) = \sum_{i=1}^{m} p_i^{\theta_A} \log \frac{p_i^{\theta_A}}{p_i^{\theta_R}}. \tag{20}$$

However, the KL divergence is unbounded and goes to infinity whenever $p_i^{\theta_A}$ vanishes

and $p_i^{\theta_R}$ does not, which requires using some kind of smoothing when estimating the distributions. Lin et al. (2006) claims that the measure used here should also be *symmetric*,[25] another thing that the KL divergence is not. Hence, they propose to use the *Jensen-Shannon divergence* which is bounded and symmetric:[26]

$$\begin{aligned} JS(p^{\theta_A}||p^{\theta_R}) &= \frac{1}{2}KL(p^{\theta_A}||r) + \frac{1}{2}KL(p^{\theta_R}||r) = \\ &= H(r) - \frac{1}{2}H(p^{\theta_A}) - \frac{1}{2}H(p^{\theta_A}), \end{aligned} \tag{21}$$

where $r = \frac{1}{2}p^{\theta_A} + \frac{1}{2}p^{\theta_R}$ is the *average distribution*.

To evaluate a summary S_A given a reference summary S_R, the authors propose to use the *negative JS divergence* between the estimates of p^{θ_A} and p^{θ_R} given the summaries,

$$\text{score}(S_A|S_R) = -JS(p^{\hat{\theta}_A}||p^{\hat{\theta}_R}) \tag{22}$$

The parameters are estimated via *a posteriori* maximization assuming a multinomial generation model for each summary (which means that they are modeled as bags-of-words) and using Dirichlet priors (the conjugate priors of the multinomial family). So:

$$\hat{\theta}_A = \arg\max_{\theta_A} p(S_A|\theta_A)p(\theta_A), \tag{23}$$

where (m being the number of distinct words, $a_1, \ldots, a_m$ being the word counts in the summary, $a_0 = \sum_{i=1}^m a_i$)

$$p(S_A|\theta_A) = \frac{\Gamma(a_0+1)}{\prod_{i=1}^m \Gamma(a_i+1)} \prod_{i=1}^m {\theta_{A,i}}^{a_i} \tag{24}$$

and

$$p(\theta_A) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^m \Gamma(\alpha_i)} \prod_{i=1}^m {\theta_{A,i}}^{\alpha_i - 1} \tag{25}$$

where α_i are hyper-parameters and $\alpha_0 = \sum_{i=1}^m \alpha_i$. After some algebra, we get

$$\hat{\theta}_{A,i} = \frac{a_i + \alpha_i - 1}{a_0 + \alpha_0 - m} \tag{26}$$

which is similar to MLE with smoothing.[27] $\hat{\theta}_R$ is estimated analogously using the reference summary S_R. Not surprisingly, if we have more than one reference summary, the MAP estimation given all summaries equals MAP estimation given their concatenation into a single summary.

[25] However, the authors do not give much support for this claim. In our view, there is no reason to require symmetry.

[26] Although this is not mentioned in (Lin et al., 2006), the Jensen-Shannon divergence also satisfies the axioms to be a *squared metric*, as shown by Endres and Schindelin (2003). It has also a plethora of properties that are presented elsewhere, but this is out of scope of this survey.

[27] In particular if $\alpha_i = 1$ it is just maximum likelihood estimation (MLE).

The authors experimented three automatic evaluation schemes (JS with smoothing, JS without smoothing, and KL divergence) against manual evaluation; the best performance was achieved by JS *without* smoothing. This is not surprising since, as seen above, the JS divergence is bounded, unlike the KL divergence, and so it does not require smoothing. Smoothing has the effect of pulling the two distributions more close to the uniform distribution.

6 Conclusion

The rate of information growth due to the World Wide Web has called for a need to develop efficient and accurate summarization systems. Although research on summarization started about 50 years ago, there is still a long trail to walk in this field. Over time, attention has drifted from summarizing scientific articles to news articles, electronic mail messages, advertisements, and blogs. Both abstractive and extractive approaches have been attempted, depending on the application at hand. Usually, abstractive summarization requires heavy machinery for language generation and is difficult to replicate or extend to broader domains. In contrast, simple extraction of sentences have produced satisfactory results in large-scale applications, specially in multi-document summarization. The recent popularity of effective newswire summarization systems confirms this claim.

This survey emphasizes extractive approaches to summarization using statistical methods. A distinction has been made between single document and multi-document summarization. Since a lot of interesting work is being done far from the mainstream research in this field, we have chosen to include a brief discussion on some methods that we found relevant to future research, even if they focus only on small details related to a general summarization process and not on building an entire summarization system.

Finally, some recent trends in automatic evaluation of summarization systems have been surveyed. The low inter-annotator agreement figures observed during manual evaluations suggest that the future of this research area heavily depends on the ability to find efficient ways of automatically evaluating these systems and on the development of measures that are objective enough to be commonly accepted by the research community.

Acknowledgements

We would like to thank Noah Smith and Einat Minkov for valuable suggestions during the course of the survey. We would also like to thank Alex Rudnicky and Mohit Kumar for insightful discussions at various points during 2006-2007.

References

Amari, S.-I. and Nagaoka, H. (2001). *Methods of Information Geometry (Translations of Mathematical Monographs)*. Oxford University Press. [20]

Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B. (1999). A trainable summarizer with knowledge acquired from robust nlp techniques. In Mani, I. and Maybury, M. T., editors, *Advances in Automatic Text Summarization*, pages 71–80. MIT Press. [4, 5]

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings ISTS'97*. [8]

Barzilay, R., McKeown, K., and Elhadad, M. (1999). Information fusion in the context of multi-document summarization. In *Proceedings of ACL '99*. [12, 13, 14, 16]

Baxendale, P. (1958). Machine-made index for technical literature - an experiment. *IBM Journal of Research Development*, 2(4):354–361. [2, 3, 5]

Brown, F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311. [18]

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA. ACM. [8]

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR '98*, pages 335–336, New York, NY, USA. [12, 14, 15]

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania. [13, 20]

Conroy, J. M. and O'leary, D. P. (2001). Text summarization via hidden markov models. In *Proceedings of SIGIR '01*, pages 406–407, New York, NY, USA. [6]

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley. [25]

Daumé III, H. and Marcu, D. (2002). A noisy-channel model for document compression. In *Proceedings of the Conference of the Association of Computational Linguistics (ACL 2002)*. [20]

Daumé III, H. and Marcu, D. (2004). A tree-position kernel for document compression. In *Proceedings of DUC2004*. [20]

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285. [2, 3, 4]

Endres, D. M. and Schindelin, J. E. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860. [26]

Evans, D. K. (2005). Similarity-based multilingual multi-document summarization. Technical Report CUCS-014-05, Columbia University. [12, 17]

Gous, A. (1999). Spherical subfamily models. [20, 21]

Hall, K. and Hofmann, T. (2000). Learning curved multinomial subfamilies for natural language processing and information retrieval. In *Proc. 17th International Conf. on Machine Learning*, pages 351–358. Morgan Kaufmann, San Francisco, CA. [20]

Hovy, E. and Lin, C. Y. (1999). Automated text summarization in summarist. In Mani, I. and Maybury, M. T., editors, *Advances in Automatic Text Summarization*, pages 81–94. MIT Press. [17]

Knight, K. and Marcu, D. (2000). Statistics-based summarization - step one: Sentence compression. In *AAAI/IAAI*, pages 703–710. [19]

Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *Proceedings SIGIR '95*, pages 68–73, New York, NY, USA. [4]

Lebanon, G. (2006). Sequential document representations and simplicial curves. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*. [22]

Lebanon, G., Mao, Y., and Dillon, J. (2007). The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, 8:2405–2441. [21, 22]

Lin, C.-Y. (1999). Training a selection function for extraction. In *Proceedings of CIKM '99*, pages 55–62, New York, NY, USA. [5]

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, S. S., editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. [8, 23, 24, 25]

Lin, C.-Y., Cao, G., Gao, J., and Nie, J.-Y. (2006). An information-theoretic approach to automatic evaluation of summaries. In *Proceedings of HLT-NAACL '06*, pages 463–470, Morristown, NJ, USA. [25, 26]

Lin, C.-Y. and Hovy, E. (1997). Identifying topics by position. In *Proceedings of the Fifth conference on Applied natural language processing*, pages 283–290, San Francisco, CA, USA. [5]

Lin, C.-Y. and Hovy, E. (2002). Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51, Morristown, NJ, USA. [23]

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165. [2, 3, 6, 8]

Mani, I. and Bloedorn, E. (1997). Multi-document summarization by graph search and matching. In *AAAI/IAAI*, pages 622–628. [15, 16]

Marcu, D. (1998a). Improving summarization through rhetorical parsing tuning. In *Proceedings of The Sixth Workshop on Very Large Corpora, pages 206-215*, pages 206–215, Montreal, Canada. [9, 10, 20]

Marcu, D. C. (1998b). *The rhetorical parsing, summarization, and generation of natural language texts*. PhD thesis, University of Toronto. Adviser-Graeme Hirst. [10]

McKeown, K., Klavans, J., Hatzivassiloglou, V., Barzilay, R., and Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI*, pages 453–460. [11, 12, 13, 14, 16]

McKeown, K. R. and Radev, D. R. (1995). Generating summaries of multiple news articles. In *Proceedings of SIGIR '95*, pages 74–82, Seattle, Washington. [8, 11, 12]

Miller, G. A. (1995). Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41. [4, 9]

Nenkova, A. (2005). Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of AAAI 2005, Pittsburgh, USA*. [7]

Ono, K., Sumita, K., and Miike, S. (1994). Abstract generation based on rhetorical structure extraction. In *Proceedings of Coling '94*, pages 344–348, Morristown, NJ, USA. [9]

Osborne, M. (2002). Using maximum entropy for sentence extraction. In *Proceedings of the ACL'02 Workshop on Automatic Summarization*, pages 1–8, Morristown, NJ, USA. [7]

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL '02*, pages 311–318, Morristown, NJ, USA. [24]

Radev, D. R., Hovy, E., and McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics.*, 28(4):399–408. [1, 2]

Radev, D. R., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 21–30, Morristown, NJ, USA. [12, 16, 17]

Radev, D. R., Jing, H., Stys, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management 40 (2004)*, 40:919–938. [16, 17]

Radev, D. R. and McKeown, K. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500. [12]

Salton, G. and Buckley, C. (1988). On the use of spreading activation methods in automatic information. In *Proceedings of SIGIR '88*, pages 147–160, New York, NY, USA. [15]

Salton, G., Wong, A., and Yang, A. C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18:229–237. [20]

Selman, B., Levesque, H. J., and Mitchell, D. G. (1992). A new method for solving hard satisfiability problems. In *AAAI*, pages 440–446. [11]

Svore, K., Vanderwende, L., and Burges, C. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. In *Proceedings of the EMNLP-CoNLL*, pages 448–457. [7, 8]

Witbrock, M. J. and Mittal, V. O. (1999). Ultra-summarization (poster abstract): a statistical approach to generating highly condensed non-extractive summaries. In *Proceedings of SIGIR '99*, pages 315–316, New York, NY, USA. [18]