Automatic Abstracting

Daniel Marcu
Information Sciences Institute and
Department of Computer Science
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA, 90292
marcu@isi.edu

May 9, 2002

Keywords: automatic abstracting, summarization

1 Introduction

After lying dormant for a few decades, the field of automated text summarization has experienced a tremendous resurgence of interest. Recently, many new algorithms and techniques have been proposed for identifying important information in single documents and document collections, and for mapping this information into grammatical, cohesive, and coherent abstracts. Since 1997, annual workshops, conferences, and large scale comparative evaluations have provided a rich environment for exchanging ideas between researchers in Asia, Europe, and North America. This article overviews the main developments in the field and provides a guiding map to those interested in understanding the strengths and weaknesses of an increasingly ubiquitous technology.
2 Genres and types of summaries

It is almost impossible to conceive of a world without summaries. Daily, we skim over headlines to decide what news to read. We go over scientific abstracts to decide what papers to study. We read reviews to decide what books to buy. And we browse TV-guides to decide what movies to watch.

Most of the summaries that we use today are produced manually, by trained abstractors. However, an increasingly large number of researchers is now tackling the problem of automatic summarization. This turns out to be quite difficult because even defining the scope of the problem is not an easy feat. Consider, for example, the text fragment in (1).

Running nose. Raging fever. Aching joints. Splitting headache. Are there any poor souls suffering from the flu this winter who haven't longed for a pill to make it all go away? Relief may be in sight. Researchers at Gilead Sciences, a pharmaceutical company in Foster City, California, reported last week in the Journal of the American Chemical Society that they have discovered a compound that can stop the influenza virus from spreading in animals. Tests on humans are set for later this year. The new compound takes a novel approach to the familiar flu virus. It targets an enzyme, called neuraminidase, that the virus needs in order to scatter copies of itself throughout the body. This enzyme acts like a pair of molecular scissors that slices through the protective mucous linings of the nose and throat. After the virus infects the cells of the respiratory system and begins replicating, neuraminidase cuts the newly formed copies free to invade other cells. By blocking this enzyme, the new compound, dubbed GS 4104, prevents the infection from spreading.

What is a good summary of text (1)? A headline such as that in (2) may convince a lay person who is hit annually by a flu to read the whole article. However, a molecular biologist interested in the function of various enzymes may be more interested in an abstract such as that shown in (3).

**Headline:** Flu stopper — A new compound is set for human testing (2)

---

1Time Magazine (02/10/1997).
**Short abstract:** The neuraminidase enzyme enables the flu virus to replicate and invade uninfected cells.

As examples (2) and (3) illustrate, the communicative goal with which a summary is written and the background knowledge of the intended audience clearly influence the information that one may consider to be important.

The intended communicative goal and the background of the audience are only two of the facets that are useful when characterizing summaries. Other facets that have received significant attention in the last couple of years are listed below. (See [1, 2, 3] for a list of dimensions along which summaries can be characterized.)

**Usage.** When they can be used only for quick categorization, summaries are called *indicative*. When they can be used as substitutes of the original documents in order to access significant content, summaries are called *informative*.

**Relation to source.** When summaries contain sentences/ clauses that are “lifted” verbatim from the input source, they are called *extracts*. When the important information in a text is paraphrased to ensure cohesion, coherence, and a higher degree of compression, summaries are called *abstracts*. A particular type of abstract, which is extremely short and uses a peculiar syntax is the *headline*. For example, the text in (4) is an extractive summary of the text in (1), the text in (2) is a headline, and the text in (3) is a short abstract.

**Coherent extract:** Researchers at Gilead Sciences, a pharmaceutical company in Foster City, California, reported last week in the Journal of the American Chemical Society that they have discovered a compound that can stop the influenza virus from spreading in animals. Tests on humans are set for later this year.

**Purpose.** When summaries provide the author’s view, they are called *generic*. When they reflect the user’s interest, often expressed in the form of a query, they are called *query-oriented*. 

3
Source type. When they are based on only one text, they are called single-document summaries. When they fuse information provided in multiple texts, they are called multi-document or multidoc summaries.

The dimensions above are not always clearly cut. A summary can be at the same time indicative and informative. Or it can be indicative for one audience and informative for another. A summary that is obtained by selecting important sentences and then simplifying them by deleting unimportant words and phrases is not really an extract, because it contains clauses/sentences that are not part of the input. But it is not an abstract either, because paraphrasing is not employed extensively. By the same token, one may argue that it is not appropriate to call “abstract” and extract in which only the pronominal references have been replaced by the corresponding noun phrases.

3 Summary Evaluation

The discussion in Section 2 is much more important than it first meets the eye. The facets and summary types enumerated above do not merely provide a common vocabulary to natural language researchers. Rather, they set the foundation for addressing a serious concern: If there are so many facets and factors that influence the way summaries are constructed and presented, how can one measure the quality of a summary? What makes a summary good and another one bad? Unless these questions can be answered adequately, it is impossible to assess progress in the field of automatic abstracting.

The first metrics for measuring the quality of a summary have been proposed in the context of producing generic extracts. Given a document, one or more human judges are asked to mark the sentences that are important. The sentences on which a majority of the judges agree to be important are taken to be the “gold standard”. The quality of automatic summarizers is measured in terms of recall — the number of important sentences correctly identified by the summarizer divided by the total number of sentences in the gold standard — and precision — the number of important sentences correctly identified by the summarizer divided by the total number of sentences selected by the summarizer. Note that both recall and precision have to be taken into consideration as one can easily rig one of the metrics. By marking as important all sentences in a text, one can easily obtain a 100% recall. Clearly though, the precision is going to be much smaller. A metric that
summarizes the recall and precision metric is the F-value metric, which is a harmonic average of precision and recall \((F = 2RP/(R + P))\). The F-value is a number between recall and precision that it is higher when recall and precision are closer.

The F-value cannot be rigged the same way recall and precision can when considered in isolation. Unfortunately, the use of the F-value metric can pose problems as well. Let’s say that five human judges mark as important sentences in a document that has 20 sentences. Let’s say that each judge marks ten sentences as important but a majority of the judges agree on only two sentences. Is this two sentence gold summary of any good? If five judges randomly choose sentences from a document and mark them as important, just by the laws of odds it is likely that two sentences will be marked as important by a majority of them. This is clearly a problem as it corresponds to a case where there is no agreement between human judges with respect to what is important in a text (See [4, 5, 6] for discussions of methods for computing agreement between multiple judges.) And if humans cannot agree on what is important, it means that a summarizer that chooses sentences randomly is as good as a human!!!

Some of the summarization studies to which I refer in this overview were carried out before statistical methods for measuring inter-annotator agreement and adequate experimental design techniques have become part of the tools used by natural language researchers. For example, a classic summarization paper published by Edmundson [7] measures the performance of various summarization methods in terms of recall with respect to a gold standard built by only one human. Fourty years after the experiment of Edmundson has been carried out, it is impossible to tell how difficult the texts given as input were. Still, as the reader will see in the rest of the overview, we still take many such results as indicative of the performance of various methods. Understanding how to measure the quality of a summary is by itself a research area [8] that evolves concurrently with research aimed at developing increasingly sophisticated summarization techniques.

The recall, precision, and F-value metrics represent only a small fraction of the metrics proposed to measure the quality of a summary. Alternative metrics have been proposed in order to address their limitations. For example, Radev et al. [9] propose an evaluation metric in which sentences are not judged on a boolean scale as important or non-important, but are given a utility score. A summarizer that selects sentences of high utility is given a higher reward than a summarizer that selects sentences of medium or low
utility.

To enable the evaluation and encourage the development of systems that produce not extracts but abstracts, the Document Understanding Conferences (DUC) [10, 11] use human judges to measure the degree of overlap between a model summary and abstracts/extracts produced by various systems. In addition to the informational content, systems participating in DUC are also evaluated in terms of output grammaticality, cohesion, and coherence.

The metrics discussed so far pertain to the intrinsic quality of a summary. However, in many scenarios, summaries are also evaluated extrinsically, with respect to their utility in solving specific tasks. For example, systems that participated in the SUMMAC evaluation [12] were evaluated by measuring how often their outputs could be used by humans to determine whether the documents for which the summaries were generated were relevant to a given topic. In another task, summarization systems were judged by measuring how often their outputs could be used to successfully carry out a document categorization task.

Reviewing all research in summary evaluation is beyond the scope of this article. The reader is referred to [12, 10, 11, 1, 2, 8] for detailed discussions of the issues involved. In the rest of this article, whenever I refer to the “performance” of a summarization system without using any additional qualifiers, I mean the F-value of that system, as measured against a gold standard. When the performance of a system was measured using a different metric, I make that explicit. Because the science of evaluating summaries is not stable and because a generally accepted evaluation standard is yet to emerge, comparing the performance of various summarization algorithms and techniques is a challenging task.

4 Summarization specific components

4.1 Overview

In dealing with the facets discussed in Section 2, researchers in summarization have developed a number of components/techniques that are now part of many summarization systems. Importance identifiers are used to determine the most important sentences and clauses in a document or collection of documents. Sentence compression algorithms are used to compress extracts
Figure 1: Components (ellipses) and sought outputs (bold boxes) of single document summarization systems.

Further, by deleting non-important phrases and clauses in long sentences. Paraphrasing components are used to increase the compression and readability of the output. Information extraction and natural language generation systems are used to generate abstracts via semantic-based representations. Headline generators use specialized techniques to determine the most representative concepts in a text and to render them using a language that is often different from that used in typical English.

Figure 1 shows the components that have received so far significant attention (the components are represented as ellipses) and the summary types these components are meant to produce (the summary outputs are represented in bold fonts surrounded by boxes). Each of the summarization pipelines in Figure 1 takes as input a document and some optional parameters, such as the desired compression, a user query, etc. Additional components that were developed to generate multi-document summaries are discussed in Section 7.
4.2 Importance identifiers

Arguably, most of the effort in the field of summarization research has gone into the development of algorithms and methods for identifying important sentences/clauses in single document texts.

4.2.1 The position-based method

Many of the current summarization systems assume that sentences that occur at the beginning of documents are more likely to be important than sentences that occur at the end. The simplest method to operationalize this assumption is to build a summarizer that always selects the first sentence in a text; or the first k sentences in a text, when a smaller degree of compression is required.

Empirical evaluations of algorithms that implement this assumption have revealed that the position-based method is the best [7, 13] or second best [14] single method for identifying important sentences in text. Although the performance of this method varies significantly with text genre and compression rate, this method is usually capable of identifying around 33% of the important sentences in a text [7, 13, 14].

An extensive study aimed at deriving a genre dependent optimum position policy have been carried out by Hovy and Lin [15] who have shown that in technical articles that announce computer related products, the sentence that is most likely to be important is the first sentence of the second paragraph, then the first sentence of the third paragraph, then the second sentence of the second paragraph, and so on; while in news articles, the sentence that is most likely to be important is the first sentence of the first paragraph, then the first sentence of the second paragraph, and so on.

4.2.2 The title-based method

Edmundson [7] was the first to show that the words in titles and headings are more likely to be used in important sentences in a text than in non-important sentences. Systems that employ this heuristic in combination with others tend to yield summaries of higher quality. For example, the title-based method increased the performance of Edmundson’s position-based summarizer by 8% and the performance of Teufel and Moens’s [14] cue-phrase-based summarizer by 3%.
4.2.3 The cue-phrase method

Cue-phrase-based systems capitalize on the observation that important sentences contain "bonus phrases" such as significantly, in this paper we show, and in conclusion, while non-important sentences contain "stigma phrases" such as hardly and impossible. The cue-phrase method yielded the best results when used to identify important sentences in scientific articles [14] — it identified 55% of the important sentences in a text — and was shown to increase the performance of summarization systems by 7 to 9% on other genres, when applied in conjunction with other methods [7, 13].

4.2.4 The word-frequency method

To my knowledge, the claim of Luhn [16, 17] that important sentences in a text are those that contain words that occur "somewhat" frequently has not been validated empirically in any summarization system. In Edmundson’s experiments [7], this method decreased the performance by 7% when combined with other methods. In Kupiec et al.’s experiments [13], this method decreased performance by 2%. The method appeared to mildly help only in Teufel and Moens’s system [14], where it increased performance by 0.2% when combined with other methods. In spite of not being properly validated, the word-frequency method continues to be used in many implemented systems.

4.2.5 Cohesion-based methods

The working hypothesis that constitutes the foundation of all cohesion-based methods is that important sentences/paragraphs are the highest connected entities in more or less elaborate semantic structures. There are several approaches to characterizing the level of connectedness. They are based on word co-occurrences, local salience and grammatical relations, co-reference, lexical similarity, and combinations of the above.

Word-based. The most straightforward approaches apply Information Retrieval techniques to compute the similarity between the paragraphs in a text [18, 19, 20]. The paragraphs that have the highest collective similarity to the other paragraphs are assumed to be central/important to the document they belong. When evaluated against a collection of encyclopedia articles, this method did not improve significantly over a position baseline [20]. The assumption that important sentences are those that contain important
concepts have been also explored more recently in a different representation space, using methods specific to latent semantic analysis [21].

**Lexical chains-based.** Lexical chains are successions of semantically related words that create a context and contribute to the continuity of meaning [22]. For example, a lexical chain defined over text (1), may contain the semantically related words {nose, fever, headache, flu, pill, ...}. Lexical chains are automatically built [23, 24, 25] using large lexical databases, such as thesauri and the Wordnet [26]. Lexical-chain-based approaches to text summarization [23, 25] assume that important sentences are those that are “traversed” by “strong” chains, where the strength of a chain is defined by its length and number of distinct words in it. Empirical evaluations have shown promising results but have not provided yet uncontroversial evidence that this summarization method yields higher performance than others.

**Connectedness-based.** The connectedness-based method [27] fuses many of the elements specific to cohesion-based methods. Systems that use this method map first the words in a text into nodes in a graph and then create arcs between the nodes whenever an adjacency, grammatical, coreference, or lexical-similarity-based relation holds between the words corresponding to the nodes. The sentences that contain the most connected words are considered important. The method has been shown to produce good results when evaluated extrinsically, on a document categorization task [12]. I am not aware of intrinsic evaluations of this approach.

**4.2.6 Discourse-based.**

The working hypothesis of the discourse-based method is that the hierarchical discourse structure of texts [28] can be used in order to determine the important sentences/ clauses in a text [29, 30, 31, 32]. Intuitively, if a paragraph $p$ in a text elaborates on a sentence $i$ in the same text, it is reasonable to assume that a summary should contain only the sentence $i$ as the information in paragraph $p$ is subsidiary to that in $i$.

Marcu [30] presents a variety of algorithms for automatically deriving the hierarchical discourse structure of texts and Marcu [30] and Carlson et al. [33] discuss methods for exploiting the discourse structure in order to determine the most important sentences/clauses in a text. For short texts,
the performance of discourse-based summarization systems approaches that of humans.

4.2.7 Integration of the methods

None of the methods above has been proven to provide consistent, high-performance results across all text genres, document lengths, and compression rates. In fact, experiments carried out by Jing et al. [34] suggest that many of these methods are quite unstable. Systems that perform well in a given summarization setting, may perform poorly in a different one. To increase robustness, current summarization systems do not rely on only one of the above methods. Rather, they use supervised machine learning techniques, such as naive bayes [13, 14], decision-trees [35, 36], inductive learning [35] and rhetorical parsing tuning [30], in order to adjust the contribution of each summarization method in determining the most important sentences in a text. In stable experimental conditions, summarization systems that use machine learning in order to integrate a variety of methods/techniques always outperform systems that use only one summarization method.

4.3 Sentence simplifiers/compressors

Professional abstractors do not create abstracts by simply copying the most important sentences in a text. Often, they delete non-important segments in the important sentences to yield summaries of higher compression [37, 38, 39]. For example, a sentence such as that in (5).a, can be simplified as shown in (5).b.

a. Although the modules themselves may be physically and/or electrically incompatible, the cable-specific jacks on them provide industry-standard connections.

b. Cable-specific jacks provide industry-standard connections.

To simplify sentences, some researchers [40, 41, 42, 43, 44, 45] have manually developed collections of rules which posit, for example, that during compression, all adjectives and relative clauses in a sentence should be dropped. As one may expect, this approach can often yield ungrammatical outputs. To address this problem, other researchers [46, 39] have tackled the sentence simplification problem in a probabilistic framework, using a noisy-channel
approach similar to that employed in speech recognition [47], machine translation [48], part-of-speech tagging [49], and information retrieval [50].

Using an automatically collected corpus of sentences and their human generated compressions, Knight and Marcu [46, 39] have developed generative probabilistic models that explain how short sentences can be rewritten as long sentences through a sequence of syntax-based stochastic operations. Mathematically, any long sentence \( L \) can be generated from a short sentence \( S \) with a probability \( P(L \mid S) \). A language model component assigns a probability \( P(S) \) to any possible sentence: grammatical sentences have high probability, while non-grammatical sentences have low probability. Once the probability distributions \( P(L \mid S) \) and \( P(S) \) are estimated, given a sentence \( L \), sentence compressions of arbitrary length can be generated by looking for sentences \( S \) that maximize the product \( P(L \mid S) \times P(S) \), i.e., sentences that are both good compressions (\( P(L \mid S) \) is high) and grammatical (\( P(S) \) is high). Empirical evaluations of Knight and Marcu’s sentence compression algorithms have shown that the noisy-channel approach outperforms two baseline algorithms that compress sentences by dropping words randomly or by choosing subsequences of maximal n-gram probability. However, Knight and Marcu’s compression system still does not come close to human performance levels.

### 4.4 Sentence paraphrasers

The simple deletion of syntactic constituents in long sentences is not the only way in which professionals create abstracts. Often, information from multiple sentences is fused together or for coherence, cohesive, and stylistic effects, information in one or multiple sentences is paraphrased. A number of researchers have studied in detail the types of paraphrase operations employed during summarization [51, 52, 38] and applied them in implemented systems [52, 38]. However, systems capable of generating multiple paraphrases of arbitrary sentences are yet to be developed.

For example, it is not clear at all what kind of computational processes one should use in order to produce the short abstract in (3), which paraphrases some of the information in text (1) while making explicit information that is not rendered in the original text. Or consider the examples of compression/paraphrasing in (6) and (7), in which two sentences have been compressed/paraphrased into one (the text in italics has been preserved or
modified slightly during compression).

1. He holds a bachelor’s degree in chemistry. (6)

2. “Maintaining an organization like ISFUG is like building a castle in the sand; it just requires constant work to keep it trim,” said Berkman, an economist with the Commerce Department’s Bureau of Economic Analysis.

Compression with paraphrasing: Berkman, who holds a bachelor’s degree in chemistry, is an economist with the US Department of Commerce.

1. Nonetheless, policies on its use vary. (7)

2. Agencies such as the Internal Revenue Service and the Farm Credit Administration promote shareware use, but one NASA center shuns it.

[Compression with paraphrasing: Federal agencies generally use shareware, but policies on its use vary.

Both examples of compression/paraphrasing in (6) and (7) have been produced by professional abstractors. Mimicking this process automatically is not within the capability of current summarization systems. Fortunately, the recognition and generation of paraphrases is currently on the radar of many research communities. Researchers in natural language generation and summarization [53], data mining and question answering [54] and information extraction [55] have recently developed methods for automatically learning paraphrase-specific information by exploiting bilingual, parallel texts [54] or large corpora of monolingual text [54, 55]. It is likely that progress in this area will have a significant impact on the next generation of automatic abstraction systems.

4.5 Coherence/coreference enforcers

Although extraction-based summaries contain, by construction, only well-formed grammatical sentences, they often do not sound right. For example, the summary extract in (8) of text (1) sounds odd because it is not clear
what entities “the new compound”, “this enzyme”, “the new compound” and “the infection” refer to. Also, even if we could felicitously solve all these references, the two sentences still do not seem to go well together because it is not clear what the discourse relation between the two is.

Incoherent extract: The new compound takes a novel approach to the familiar flu virus. By blocking this enzyme, the new compound, dubbed GS 4104, prevents the infection from spreading.

A number of techniques have been proposed to rewrite extracts so as to yield more readable summaries. Many of the systems tested as part of the large scale summary evaluations carried out in conjunction with the Document Understanding Conferences [10, 11] use anaphora resolution techniques in order to replace with name entities some of the pronouns in the extracts they produce. More sophisticated summary revision operations, such as aggregation of constituents of two sentences on the basis of referential identity and reduction of co-ordinated constituents have been proposed by Mani et al. [56]. Aggregation operations specific to events have been proposed by Maybury [57]. And sentence-reordering algorithms have been proposed by Barzilay et al. [58].

In an attempt to deal with sentence and document compression in a uniform framework, Daumé and Marcu [59, 60] have shown how the noisy-channel approach to sentence compression of Knight and Marcu [39] can be extended to the text level. Their empirical evaluations show that a document compression system that integrates the deletion of syntactic constituents with the deletion of clauses and sentences outperforms a compression system that sequentially simplifies the sentences in a text.

5 Headline generators

The vast majority of the work on headline generation has been carried out in a statistical-based noisy-channel framework [61, 62, 63, 64]. Given large collections of (text, headline) tuples, which can be easily collected from the web, one can estimate probability distributions $P(w_d \mid w_h)$ that reflect the likelihood of a word $w_d$ occurring in a document when another word $w_h$ occurs in a headline. By treating documents and headlines as bags of words, one can easily estimate the probability $P(D \mid H)$ of a document given a headline. A
classic trigram language model $P(H)$ can be trained to differentiate between well- and ill-formed headlines. Once the parameters of these models are estimated, one can construct document headlines by searching for sequences of words $H$ that maximize the product $P(D \mid H) \times P(H)$. The headlines generated in this manner tend to contain the most representative words in a document; however, the n-gram language models that were employed so far in headline generators seem to be too weak to enforce that the generated headlines are grammatical.

A different approach to headline generation is taken by Daumé et al. [65] who first identify the input document type. This type can fall into one of these four classes: single event, multiple event, biography, and natural disaster. In addition to the type, Daumé et al. also automatically determine the most salient entities and relations in the input. Depending on the input type, they use a set of predefined templates in order to produce grammatical headlines. For example, when a multi-event template of the form “MainEvents in Location1, Location2, and Location3” is instantiated using entities specific to a collection of documents about eclipses, it yields the headline “Eclipses in Hawaii, Mexico, and Bay Area”. The method produces impressive results when the document type and salient entity identifiers work correctly; but it produces odd results when these components fail.

6 Summarization via semantic representations

Most of the light semantic-based summarization systems developed to date implement the pipeline architecture in Figure 1. An information extraction system specialized in dealing with certain events, such as “terrorism” or “natural disasters” first extracts from the document(s) to be summarized text fragments that fit a predefined template (see Figure 2 for an example). Once the template has been filled, a natural language generation component is used to map the information in the template into well-formed natural language sentences [66, 67].

Since the types of entries associated with information extraction templates are limited, before generation, one can apply additional operations to modify, delete, and aggregate information extracted from text [67]. This is particularly useful in the context of multidocument summarization, where the information extracted from multiple articles may be inconsistent.

This approach to summarization produces excellent results in the limited
domains for which the information extraction and natural language generation systems have been tuned. Unfortunately, the method does not generalize easily to arbitrary domains. In general, the role of semantics in summarization is still to be determined. What constitutes an appropriate, computable representation for the semantics of texts that can be exploited in the context of summarization applications remains a controversial issue. Some researchers [68] map texts into description logics and perform condensation operations on formal representations. Others [69, 66, 65] tend to use intermediary representations that are closer to the surface and/or syntactic forms. No researcher in the field claims that high-performance summarization systems can be built without a deeper understanding of the semantics of texts than current technology supports. But where the middle ground between formal elegance and coverage is remains an open question.

7 Multidoc summarization

A multidocument summarization system takes as input a collection of documents and produces a summary of the entire collection. In some instances, producing generic multidocument summaries is easier than producing generic single document summaries. If a collection of documents is about the same event, let’s say an earthquake, it is likely that many of the input documents contain sentences describing the number of victims, location, time, and strength of the earthquake. Clustering techniques that measure the word overlap between sentences are usually sufficient for choosing from the set of all sentences in a collection one that is similar to many others. And if a sentence and its variants was produced by many authors, it is likely to be important. In such instances, important information can be hence detected using simple word overlap. In other instances, producing multidocument summaries
is more difficult than producing single document summaries. When the collection given as input is heterogeneous and describes multiple events of the same kind, for example earthquakes that happened in a variety of locations over a time interval, it is no longer easy to determine what information is most important.

Besides the challenges specific to single document summarization, multidocument summarizers have to address a whole range of additional problems. When one summarizes document collections, it often happens that the information given in them is inconsistent. One document may claim that a car crash has produced six victims, while another document may claim seven. Which document should one believe? And how can one determine that the two documents talk about the same car crash, to start with?

Determining the absolute time of the events reported in the input documents becomes an important factor in deciding what should go in a summary. If a collection reports the saga of a military conflict, what should go in a very short summary? The cause of the conflict? The major steps in the conflict? Or only the outcome?

Equally important is the explicit mentioning of the dates/times at which the events have occurred [70]. If the dates/times associated with the reported events are not explicitly mentioned, there is a serious danger for human readers to make incorrect inferences. The simple juxtaposition of some sentences may mislead readers to believe that, for example, an event in one sentence have caused the event in the subsequent sentence if the reader cannot infer from the summary that the events took place ten years apart.

Collections about similar events tend to contain significant amounts of redundant information. But recognizing where redundancies occur is by no means a trivial problem as the relation between sentences in arbitrary documents is difficult to characterize [71]: sentences can be semantically equivalent, they can subsume each other, or the information they convey can partially overlap. Recognizing the exact relation that holds between two arbitrary sentences is still an open problem.

The field of multidocument summarization has created the context and need for research in a variety of areas. It led to the development of methods that are well suited for generating non-redundant summaries [72, 73]; methods that can be used to detect and track events over time while producing “evolving”, temporal summaries [74]; methods for fusing sentence fragments [69, 66] in order to produce grammatical sentences; and in-depth studies of specific summary types (biographical) [2].
Since 2001, the Document Understanding Conference [10, 11] in North America and the NTCIR Workshop on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization [75] in Asia have provided an excellent forum for evaluating and comparing a variety of multidocument summarization algorithms. Participants in large scale evaluations worry not only about building useful systems but also about figuring out how to measure their performance [76, 77]. At the moment, it appears that there exists a certain gap between the summarization systems that participate in DUC and NTCIR evaluations, which are capable to handle unrestricted document collections using simple surface-based techniques, and the systems developed to advance the state of the art in information fusion [69, 66], paraphrasing [53], or text compression [59]. It would be interesting to see how and when the latter work gets integrated into robust summarization systems capable of handling arbitrary text types.

8 Other kinds of summaries

Throughout this article, I have focused primarily on techniques and algorithms developed for producing single and multidocument generic summaries. By no means this covers all the work in text summarization. For example, Teufel and Moens [78] go beyond generic summarization and devise algorithms for classifying sentences according to their rhetorical goal (Background, Topic/Aboutness, Related Work, Purpose/Problem, Solution/Method, Result, and Conclusion/Claim). A system capable of recognizing these roles can be used to generate summaries tailored to specific communicative goals.

An increasingly significant body of work has been focusing on summarizing dialogues and meeting transcripts [79, 80, 81], video [82, 83], diagrams [84], soccer games [85], and web pages for PDA access [86]. As we make progress in developing new methods for representing and accessing non-textual media types, it is likely that we will develop additional techniques for summarizing non-textual information that exploit peculiarities specific to various media types. For example, Amitay and Paris [87] propose a method for summarizing web pages that makes use of the text in the links that point to a page.
9 Conclusion

The field of natural language summarization has come a long way since the initial experiments of Luhn [16]. Although we still struggle to define what summaries are and attempt to figure out how to evaluate them, there is much reason for optimism. Summarization systems are no longer the appanage of a small research community. Summarization technology is used daily to summarize news (www.cnn.com); to consolidate and summarize news published by multiple sources (www.cs.columbia.edu/nlp/newsblaster/); to provide hand-held device access to information (www-diglib.stanford.edu/testbed/doc2/PowerBrowsing/index.html); to enable voice-based navigation of the web (www.voxera.com); and to provide feedback to students on their written essays (www.etsttechnologies.com). Given the increasing success of the technology, there is a certain danger to believe that summarization is a solved problem. It is certainly not. The current success of the technology is not determined so much by the quality of the output as by the willingness of the users to put up with anything that enables them to deal easier with the deluge of information that they are subjected to. As the expectations of the users increase, the need for developing new techniques for automatic abstracting will increase too. Building a summarization system that is better than a dumb one that selects the first n sentences in a news article is still a significant challenge.

References


20


[24] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane


