User-Sensitive Text Summarization: Application to the Medical Domain

Noemie Elhadad

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2006

ABSTRACT

User-Sensitive Text Summarization: Application to the Medical Domain

Noemie Elhadad

In this thesis, we present a user-sensitive approach to text summarization. One domain which would highly benefit from tailoring summaries to both individual and class-based user characteristics is the medical domain, where physicians and patients access similar information, each with their own needs and abilities. Our framework is a medical digital library for physicians and patients. We describe a summarizer, which generates summaries of findings in an input set of clinical studies. When a physician is treating a specific patient, he's looking for information relevant to the patient's history and problems. The summarizer takes the user's interests into account and presents only the findings pertaining to a user model, as approximated by an existing patient record. The same synthesis of information can also be of interest to the patient. The summarizer predicts which medical terms used in a text will be too technical for patients, and augments it with appropriate definitions when necessary.

We adopt a generation-like architecture for our summarizer. However, because our input is textual and not semantic, new challenges arise. We operate over a content representation hybrid between full-semantic and extracted phrases. Our content organization strategy is dynamic and data-driven. This is in contrast to most summarizers which use no explicit strategies to order information extracted from several input documents. The result is more readable, coherent output. To generate the actual summary, the summarizer makes use of aggregation and phrasal generation. The result is a concise and fluent summary.

One key challenge when it comes to adapting a text for a different audience is identifying the bottleneck for reader comprehension. We analyzed corpora of technical and lay medical texts and qualified differences. We identified the presence of difficult vocabulary as the major obstacle to comprehension for lay readers. We designed an unsupervised method to predict which terms are incomprehensible for lay readers and provide the user with appropriate definitions.

Our methods are grounded on corpus analyses and feasibility studies conducted with physicians and consumers of health information. To assess the value of our work, we evaluated our summarizer both intrinsically and extrinsically. Our task-based evaluation conducted with physicians at the ICU demonstrates that personalized summaries help physicians access relevant information better than generic summaries. Evaluation with lay readers shows that our method to augment technical medical texts improves readers' comprehension significantly.

Contents

List of Figures

List of Tables

To Alex

–SBSN, a.k.a. MN

Chapter 1

Introduction

Text summarization has become increasingly established over the past ten years, paving the way for readily accessible robust summarizers. Users of text summarization are many and range from Internet surfers lacking the time to locate and digest all the latest news available on the web to scientists unable to keep pace with the burgeoning number of technical publications who must, nonetheless, be familiar with the latest findings in their fields. But research in summarization so far has been premised upon certain assumptions that are both enabling and simultaneously limiting. One of these concerns the role of the user in influencing the content and wording of a summary.

Can a summarizer be sensitive to the user? In *The Mirror and the Lamp* (1953), M. H. Abrams' classic study of trends in literary criticism, he describes a literary work at the center of a triangle consisting of the "author," the "reader," and the "universe." Most theories of literature have juggled these four major elements (work, author, reader, and universe), while tending to privilege one. Until the early nineteenth century, literary theory dealt largely with the literary work's relation-

ship with the universe (and its relationship with the reader from a purely didactic standpoint). In "mimetic theory," as Abrams called the work-universe axis, the value of the literary work is in its ability to represent the world. With the dawn of the Romantic era, the role of the author gained prominence, and interpreting the literary work became a matter of discerning the author's intention. Abrams calls this approach "expressive theory." In the twentieth century, the focus of critics turned toward the work as an object in isolation. This approach, "formalist theory," brought the work and its features to the fore, while exiling its relationship with reader, author and universe to the periphery. Post-dating the publication of Abrams' study, in the early seventies, the importance of the relationship between the reader and the work re-emerged in the form of "Reader-Response theory," which emphasizes the way the reader understands the text (Eco, 1979). In this approach, the text's meaning does not reside solely in the work by itself, the author's intention, or in the work's relationship to the universe. Rather, the critical focus is on identifying the manner in which the meaning of the text is constructed through the interaction between the text and its reader.

Similarly, research in Natural Language Processing has, over time, variously emphasized these relationships among text, reader, world, and author. While research in understanding, and especially the symbolic AI approach, has focused on the relationship between the author and the text, generation systems have relied primarily on the relationships between the world, the text, and the reader.

Examples of generation systems that adopt a mimetic approach include those which generate weather reports or stock-market reports. In these, there is no explicit goal other than representing the world, as defined by a knowledge base of

facts, to inform a generic consumer of information. Accordingly, a good report is one that represents the world accurately. The reader does not play any role in the construction of the text.

In the realm of summarization, the formalist approach has thus far been prevalent. Unlike systems that adopt a mimetic approach to generation, summarization systems do not set out to represent the world; rather their primary goal is to represent text. Moreover, most summarizers generate summaries using a generic notion of salience. In other words, what is important to summarize is determined by features of the text only, not by what the author intended or why the reader is reading the text. Whether the features of the text to summarize are inherent to it or are determined by comparison to other texts, the sole focus remains the features of text.

The Reader-Response approach to generation has also been investigated, mostly in generation of argumentative texts. Information about the reader is used to construct a more convincing text, and thus, the reader actively participates in the generation process. In such systems, the reader is represented from a cognitive standpoint; that is, the system relies on a representation of the reader's "internal" state.

In this thesis, we pose the question of whether it is possible to adopt a Reader-Response approach towards text summarization. Research in psycholinguistics and computational linguistics has long acknowledged that the reader plays, and should play, an active role in constructing the meaning of a summary (Kintsch, 1988; Edwards and Smith, 1996; Gerrig, Kuczmarski, and Brennan, 1999; Spärck-Jones, 1999): summarization is not only a function of the input text(s) but also

of the reader – who the readers are, what their knowledge before reading the summary consists of, and why they want to know about the input texts. However, both communities agree that trying to model the reader's internal state is far too complicated, if not entirely impossible. This may explain why, given this dilemma, researchers have relaxed the problem and assumed a formalist approach in order to focus on the primary task at hand: identifying the features of a text that define salience.

Here, we investigate whether it is possible to incorporate the reader in the summarization process without going inside his mind, as generation systems of argumentative texts attempted to do. Instead, we compromise with formalism by tailoring text to the reader based on the texts that we already know the reader understands or is interested in. Acquiring such a user model is by itself a subject of research and is outside the focus of this work. Rather, we assume the existence of a user model and focus on the generation of user-sensitive summaries.

Two types of user tailoring are examined in this thesis: *individualized*, i.e., what the reader is interested in, and *class-based*, i.e., the degree of expertise of the reader. Our research framework consists of a digital library that provides tailored access to medical literature for both physicians and patients. When treating a specific patient, physicians will want to keep abreast of the latest findings that pertain to their patient. Likewise, patients may want to access the latest findings that are relevant to their medical situation but may be hindered by the jargon commonly used in technical medical texts. TAS (Technical Article Summarizer), our summarizer developed as a testbed for our research, attempts to provide both types of users with tailored syntheses of the findings in clinical studies. Such tailoring

is accomplished at the *individual* level by taking advantage of the existing patient record in the digital library. The summarizer also adapts the summary at the *class* level based on examples of texts that are likely to be understood by the user, depending on whether the user is a physician or a patient.

1.1 TAS: a Multi-document User-sensitive Summarizer

We implemented the user-sensitive multi-document summarizer TAS as part of the larger project PERSIVAL to test the methods described in this thesis. TAS operates over the medical domain and relies on an existing user model consisting of an electronic patient record.

1.1.1 The PERSIVAL Project

PERSIVAL (Personalized Search and Summarization over Multimedia Information) is designed to provide tailored access to a distributed digital library of multimedia medical literature. It focuses on cardiology patients and is an interdisciplinary project that involves researchers in computer science, electrical engineering, medical informatics, and library and information science.

A key feature of PERSIVAL is the ability to present information relevant to the user's query given the context of patient information. PERSIVAL links to the large online patient record database available at the New York Presbyterian Hospital, which serves as part of the user model (Hripcsak, Cimino, and Sengupta, 1999). Interaction with PERSIVAL begins with access to a specific patient record.

After viewing the patient record, the user may decide to access the online medical literature and pose a question in natural language. The Query Formulation module helps the user to formulate a good question related to the patient information (Mendonca et al., 2001) and translates the natural language question into a query. The query is then sent to a search engine, which allows access to distributed online textual resources (Green, Ipeirotis, and Gravano, 2001), as well as a library of digital echocardiograms. The results of the text search are re-ranked by matching the articles returned against the patient record, scoring those articles which discuss results related to the patient's case as more relevant (Teufel et al., 2001). A text summarizer (Elhadad et al., 2005) and a video summarizer (Ebadollahi et al., 2001) each generate a summary of the relevant results. The resulting multimedia summary and search results are presented using a sophisticated layout component (Lok and Feiner, 2002).

Depending on both the user's expertise and the type of question asked by the user, PERSIVAL invokes different summarizers. For users who want to get overviews about specific diseases, Centrifuser (Kan, 2003) provides indicative summaries of consumer health articles. For users who want to know about the findings in the technical literature, TAS, described in this thesis, produces briefings of clinical studies that are relevant to the patient in question. In addition, TAS adapts the summary to the level of expertise of the user. When the user is a physician, then the language used in the clinical studies is appropriate; when the user is a lay person (e.g., the actual patient), then TAS adapts the text to match better the user's understanding level of medical texts.

1.1.2 Input Characteristics

TAS relies on an existing static user model. Two types of data are encoded: who the user is (physician vs. lay user) and patient information. When the user is a physician, this information refers to a patient under care, while for a lay user, it will most likely point to the user's own patient information. Patient information, in our framework, comes from a pre-existing patient record. The online patient record contains information collected over time by physicians and nurses and, thus, our user model is obtained in a non-intrusive way, for free.

The types of questions TAS is called upon to answer are open-ended (for instance, "What is the best treatment for atrial fibrillation given this patient?"). Input articles do not explicitly answer such questions; rather a set of findings is presented to the reader. Similarly, TAS does not aim to provide an explicit answer to the user's question. The summaries function as an access point to the results of the study or studies that are relevant to the patient and let the users infer their own answer. To organize the summary content, the terms used in the query are taken into account to increase/decrease the importance of individual pieces of information.

The set of documents to be summarized is the result of a search, given the user query, on the digital library restricted to medical journals. However, there are many different types of publications within medical journals (e.g., letters to the editor, case reports, reviews, and clinical studies). Based on the results of initial user studies conducted to gather TAS specifications, TAS is restricted to summarizing clinical studies. To ensure that the documents fed to TAS will only be clinical studies, articles that do not fall into this category are filtered out automatically

Aggressive lipid-lowering strategy and moderate low-density LDL-C lowering strategy were associated with atherosclerosis progression [1,2].
Predictors for atherosclerosis progression and graft worsening were stenosis of the graft, prior myocardial infarction, years post CABG, high triglyceride level, small minimum graft diameter, low HDL-C, high LDL-C, high mean arterial pressure, low ejection fraction, male gender, and current smoking [1,1].
There was no association between warfarin and progression of atherosclerosis [2,4].
Predictors of late MACE were unstable angina and CHF [3,3].

Figure 1.1: Summary example.

using a classifier. Once the clinical studies are identified, their main clinical task (i.e., prognosis, treatment, or diagnosis) is identified (see Appendix B for details on the two classifications).

1.1.3 Output Characteristics

The summaries produced by TAS are briefings presenting results reported in clinical studies (as opposed to the patient group descriptions, methods or discussion of the studies). Figure 1.1 shows an example of a summary output generated by TAS for a physician treating a specific patient (the full scenario consisting of patient information and input articles is available in Appendix A). The results included in the summaries are ones that directly pertain to the patient information stored in the user model. In this example, the patient has hypertension, high cholesterol, and a history of smoking. As such, "male gender" and "current smoking" are examples of characteristics that pertain to the patient.

The summary does not contain repetitive information (repetitions are identi-

Sex and age did not predict varicella. Varicella was associated with aspirin use and coumadin use.

Figure 1.2: Two summary sentences for a lay reader.

fied and fused) and signals to the reader any possible contradictory results found in the input articles. More generally, the summary content is organized in a dynamic fashion, presenting together pieces of information that relate to each other. In the first sentence in Figure 1.1, the information is drawn from two different sentences in article 1 and merged into one sentence. The numbers in brackets refer to the input articles. In the PERSIVAL interface, they are links to the actual sentences from which the given summary content has been drawn. The last sentence in the summary, for instance, contains references to two separate sentences in the same article 3. Finally, the summary phrases are re-used from the extracted information from the input articles.

The summary presented to a patient is similar to the summary generated for physicians. The text is augmented with definitions when appropriate. Figure 1.2 shows a summary sentence targeted at a lay reader. The two terms "varicella" and "coumadin" were determined too complex for a lay audience, and a definition is provided for each as a link (varicella is defined as "Chickenpox" and coumadin as "A blood-thinning drug"). The remaining terms (sex, age, aspirin) are classified as familiar words and thus are not defined. The summary adaptation takes as input a fully generated text, not a semantic representation. This way, the same algorithm can be applied to human-written text judged too complex for lay readers.

1.1.4 Architecture

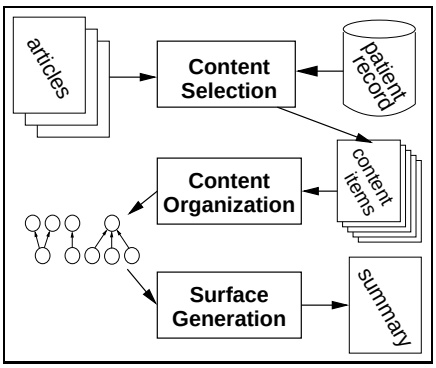


Figure 1.3: TAS architecture.

TAS follows the pipeline architecture shown in Figure 1.3. Following a generation framework, it contains three main stages: content selection, content organization and surface realization. Starting at the content selection stage, different pieces of information called *content items* are instantiated for each input article (Content Extraction). A content item is a shallow semantic representation of a candidate piece of information to include in the summary. The content items are collected from the input articles using information extraction methods. The content items that do not pertain to the patient information stored in the user model are filtered out (Content Personalization). Personalization is achieved by applying a set of strategies that operate over content items.

During the Content Organization stage, the relevant content items are clustered into semantically related clusters. Each cluster represents a paragraph-to-be in the summary, and repetitions and contradictions are identified at this stage (Merging). Each cluster is assigned a priority weight based on internal features such as number of related content items or number of repetitions, on the premise that

content items appearing in several input articles are more likely to contain high-priority information. The priorities allow the clusters to be ordered in a bottom-up fashion (Ordering).

Finally, the internal structure is realized into an English text using phrasal generation (Surface Generation). Each cluster is translated into a paragraph. Sentence planning, and aggregation in particular, determines how many sentences each paragraph contains. The realization stage does not rely on a pre-defined lexical chooser, but rather, re-uses the actual phrases stored at the content selection stage. This way, the generated summary resembles the input articles in style and wording. When the user is a patient, however, a post-processing step occurs where noun phrases are supplemented with a definition when needed so that the resulting summary is less technical and more comprehensible to the user. The decision to define a term relies on corpus analysis of medical texts targeted at health consumers and corpora of general English. The definitions themselves are selected from the Google "define:" functionality.

1.2 Contributions

The main contribution of this work is a strategy for **generating user-sensitive summaries**. There are two principal axes of the research: incorporating personalization into the summarization process and adapting a generation framework to the design of a summarizer. From a functional viewpoint, this work presents a summarizer that synthesizes information from technical medical texts.

1.2.1 Personalized Summarization

We present a summarizer that takes the reader into account at two levels: individual and class-based.

- For individual personalization, our user model relies on the existing patient record of the patient under care. As such, the user model contains a large quantity of information obtained in a non-intrusive way. Because we aim to provide a robust summarizer, we choose not to use inferences or any complex reasoning to improve the quality of the user model. This work investigates the extent to which it is possible to rely on an existing user model. In particular, we investigate how the user model affects the content of the summary.

- At the class level, we simplify a summary originally generated for physicians and written in technical language, so that a lay reader has a better chance of understanding it. Our experiments confirm that medical terminology is a major obstacle to comprehension for lay readers. Relying on examples of texts we know lay readers understand, we determine which terms likely hinder comprehension and provide appropriate definitions.

1.2.2 Summarization as a Generation Process

We treat the summarization process as a text generation process, with a content selection step, followed by content organization and surface realization. Our decision to approach summarization from a generation standpoint affords multiple advantages and opportunities but also requires us to remain cognizant of the respects in which summarization presents its own unique challenges.

- Text generation is an established field of natural language processing. Thus, operating in a generation framework allows us to build upon relevant previous research, especially when it comes to generation systems that take the reader into account. However, because summarization is, after all, different from generating a text from a non-linguistic knowledge base, each module was modified to address the challenges of summarization, as will be shown in the upcoming chapters.

- Our research on content selection identifies an appropriate semantic level at which to extract salient information without sacrificing robustness. Our representation, which we call a Content Item, is built automatically from extracted phrases and associated semantic roles.

- The strategy we developed for content organization is bottom-up: the search for an appropriate order is driven by the characteristics and similarities of the extracted pieces of information, thereby avoiding the strict top-down ordering policies characteristic of traditional generation systems.

- Finally, class-based personalization is modeled as a text-to-text generation problem, where the input is not a semantic representation like in traditional generation systems, but a sentence. As such, our method can be applied to already existing technical texts, like clinical studies.

1.2.3 NLP in the Medical Domain

Our work contributes to research in natural language processing in the medical domain.

- Because most of the basic tools for NLP (part-of-speech taggers and syntactic parsers) were developed and tested for general English, processing technical medical texts presents its own challenges. This work provides basic tools to process medical texts efficiently and accurately.

- Beyond addressing the challenges entailed in the implementation of NLP systems in the medical domain, our summarizer is a helpful tool for physicians to navigate the sea of information they confront on a daily basis in a more efficient fashion.

- Finally, our work contributes to the ongoing research on health literacy by helping patients and non-medically knowledgeable readers have better and easier access to medical literature.

1.3 Guide to Remaining Chapters

Chapter 2 presents previous work in text summarization, text generation, and, more specifically, generation/summarization with respect to personalization. While Chapter 2 presents an overview of relevant applications, each subsequent chapter provides a discussion of related techniques in more detail.

The next chapters follow the summarizer's architecture. In addition to a review of previous work for each module, intrinsic evaluation results are provided.

Chapter 3 describes the content selection module and how personalization with respect to the reader's individual points of interest is handled. We also define the atomic objects involved in the summarizer's algorithm.

Chapter 4 presents the data-driven merging and ordering strategy. The method we describe is inspired by an analysis of manual orderings. The ordering strategy was applied to a news summarizer in addition to the summarizer presented here. We report results for both summarizers to indicate that our strategy is easily adaptable to summarizers serving widely divergent functions.

Chapter 5 focuses on the sentence generation function (aggregation and surface realization). Because the realization function re-uses phrases from the input texts, the resulting sentences are targeted at physicians and medically knowledgeable readers.

Chapter 6 presents the evaluation of the summarizer as a whole for physicians. We conducted a user study to determine the added value of a user-sensitive summarizer. Physicians were tested in a task-based setting under different scenarios.

Chapter 7 is concerned with the adaptation of technical sentences for lay readers. In the summarizer, the adaptation is applied as a post-processing editing of the generated summary. This chapter presents the new data used for the adaptation, as well as a feasibility study. We describe our two-step method to augment sentences with background knowledge.

Finally, Chapter 8 discusses the limitations of this work and future work in the direction of personalization, and ends with conclusions.

Chapter 2

Related Work

In this chapter we report on related work in three broad categories: (1) text summarization, (2) user modeling with respect to text generation and summarization, and (3) text-to-text generation and text simplification. We focus here on an overview of applications with respect to these categories. We provide a more specific review of related techniques in each subsequent chapter.

2.1 Text Summarization

Text summarization is now an established field of natural language processing, attracting many researchers (DUC 2002-2005). Summarization systems can be classified according to several criteria: what type of information they provide (indicative vs. informative), what their generation strategy is (from a pure concatenation of extracted sentences, to an extract with local modification, to full content regeneration), whether they are restricted to a specific domain or are able to process texts from any domain, and whether they are genre-dependent or not.

Research in the field started with summarizing single documents. One effective technique is to extract sentences. Extractive summarizers can rely on shallow features of sentences, such as word frequency, presence of keywords, and sentence position, or a learned combination of such features (Luhn, 1958; Edmundson, 1969; Paice and Jones, 1993; Kupiec, Pedersen, and Chen, 1995).

Because some sentences in a text can be very relevant, while containing irrelevant information, researchers have investigated ways to identify and extract important clauses or phrases. Rhetorical Structure Theory has been shown helpful to identify salient clauses within sentences (Marcu, 1997). Other systems have focused on the task of text compression. One example of work on statistical text compression is automatic title generation. Existing methods rely on corpora of documents and their associated titles (Witbrock and Mittal, 1999; M. Banko, 2000). Grefenstette (1998) provides a method to produce telegraphic style summaries based on the syntactic features of input sentences. Sentence compression tackles the problem of removing irrelevant spans of words from a given sentence. Knight and Marcu (2000) propose a noisy-channel model for sentence compression while Cut and Paste summarization provides operators drawn from an analysis of human written abstracts (Jing and McKeown, 2000). The task of sentence compression is in fact a good example of text-to-text applications, as discussed at the end of this chapter.

When it comes to multi-document summarization, new challenges arise. It is necessary to identify similarities across the input documents, as well as differences, in the form of new or contradictory information. Depending on the degree of semantics involved in the content representation, similarities and differences can be detected with larger or smaller granularity. Approaches range from representing

the summary content as templates and comparing them directly (Radev and McKeown, 1998) to treating sentences as vectors of words and comparing them through similarity metrics such as cosine similarity (Goldstein et al., 1999; Radev, Jing, and Budzikowska, 2000) or more complex metrics relying on lexical and syntactic features (Hatzivassiloglou et al., 2001; Schiffman, 2005), to treating words in the input texts and their relations as a graph to identify similarities and differences. Our work builds on knowledge-rich approaches to select the summary content. In Chapter 3 we explain how our approach departs from previous work.

While some multi-document summarizers opt for a sentence extraction scheme (Goldstein et al., 1999; Mani and Bloedorn, 1999; Lin and Hovy, 2002; Schiffman, 2005), others have examined ways to combine information from different input texts into one sentence. SUMMONs (Radev and McKeown, 1998) shows that it is possible to synthesize information from different sources into sentences when relying on a deep semantic representation of the input. Information fusion (Barzilay and McKeown, 2005), on the other hand, investigates how to fuse information from multiple sentences into a single sentence without relying on a deep semantic input. Both approaches do not generate the summary content from scratch, but rather re-use phrases from the input articles. Our work follows this strategy. In Chapters 4 and 5 we describe how our methods differ from other language re-use generation approaches.

Most summarizers have been developed for the news genre and contain domain-independent summarization strategies, but there has also been significant work on domain-dependent summarization and summarization of texts from different genres. The summarization of scientific articles is an attractive application, as

there is an overwhelming number of scientific papers published every day, and it is harder and harder for researchers to keep abreast of all the latest developments relevant to their field. Paice and Jones (1993) used information extraction techniques and template-based generation to provide focused summaries of scientific papers. Kupiec, Pedersen, and Chen (1995) trained a classifier based on documents and abstracts to learn to extract sentences. They focused on scientific engineering articles. Argumentative zoning (Teufel and Moens, 2002) was also developed for scientific articles. Using supervised learning, this approach assigns a rhetorical class to each sentence in a scientific article and provides an extractive summary. This way, depending on the type of summary desired (one that focuses on results vs. one that focuses on citations, for instance), the summarizer can adjust accordingly and extract relevant sentences. Our summarizer also operates over scientific articles, but the main difference with these systems is that it summarizes multiple articles. In addition, our summarizer does not produce a generic summary, but relies on a user model to select and generate user-sensitive summaries.

2.2 User Modeling in Text Summarization and Generation

The user modeling research community is very active, whether on methods to acquire user models or on methods to use the models to customize applications (UM, 1994-2005). User modeling and natural language processing applications have some areas of overlap that have been investigated in generation, understanding and dialogue systems (for a detailed review, see Zukerman and Litman (2001)).

One of the first generation systems to take user characteristics into account was developed as part of the MYCIN expert system (Wallis and Shortliffe, 1985). Steps in the explanations generated by MYCIN are assigned a complexity and an importance weight, and users have a level of expertise encoded. Steps can be skipped if the user is an expert, or details can be omitted if the user is less knowledgeable. Content planning is the only module affected by the user modeling. This is the case in most user-sensitive generation systems. Other applications where user modeling is carried out at the content planning stage only include the Alfresco project (Stock and the ALFRESCO Project Team, 1993), and the ILEX system (Milosavljevic and Oberlander, 1998).

Using the tutorial system TAILOR and an explicit user model as a framework, Paris (1993) argues that, in the generation of explanations to users with different levels of expertise, most of the stages in the generation pipeline should be affected, especially the content selection and the content organization phases. In the same line of work for dialogue systems targeted at users with different levels of expertise, Bateman and Paris (1989) show that the sentence planning stage can also take the user into account and help generate utterances appropriate to each type of user. More recently, the SkillSum project (Williams and Reiter, 2005) has focused on adapting texts for users with poor-reading skills. The project's emphasis is on the micro-planning level, to determine which connectives are more appropriate given the user's literacy level.

The Pauline system (Hovy, 1988) relies on user characteristics beyond level of expertise, viz., the user's emotional state, as well as his/her political opinions. Given the same set of facts, different descriptions of one event can be generated

depending on the user's viewpoint. Both the content and the wording are affected by this implicit user model.

Our work departs from these systems in that it is not a deep generation system: Our input is textual, as opposed to a knowledge base of facts, and we do not want to generate text from scratch, but rather reuse phrases from the input texts. When it comes to tailoring – as we will see in Chapter 3 and 7 – the only steps of our system affected by the user occur at the content selection and realization stage. Moreover, while most generation systems would tailor their wording at the same time as they generate it from a semantic representation, our method is in two passes: first generate, then tailor the wording.

One domain that clearly benefits from tailoring the output to the user's characteristics is the medical domain. Many generation systems have been developed, typically relying on existing patient information to encode both class-based and individual characteristics of the users (whether patients or physicians) (Carenini, Mittal, and Moore, 1994; Osman et al., 1994; Binsted, Cawsey, and Jones, 1995; Cawsey, Jones, and Pearson, 2000; Lennox et al., 2001). In work to date in the medical domain, the user model affects only the content planning stage. Several domain-dependent content plans are pre-written; depending on the user characteristics, a specific plan gets instantiated. Our work is similar to these systems as our user model consists of an existing patient record. However, the internals of our system differ from these as it is not a deep generation system in nature, but a text summarizer dealing with shallow semantics.

In text summarization, while most text summarizers to date have no knowledge about the user's interests or characteristics, some summarizers are query-

based: that is, they allow the user to specify a few keywords as important in the form of a query and take this information into account when selecting summary content (Saggion, Bontcheva, and Cunningham, 2003; Kan, 2003). More recently, the Document Understanding Conference (DUC 2005) included the need for user modeling in its guidelines, requiring competing systems to have user and task or context information available to the summarizers.

We are aware of only one summarization system, aside from that described in this thesis, that tailors its output to the user based on an explicit user model. SumItBMT (Becher, Endres-Niggemeyer, and Fichtner, 2002) is a multi-document summarizer of medical articles for physicians (the prototype focuses on articles about bone marrow transplantation). The SumItBMT user model is acquired at the beginning of each session by asking the physician to fill out a form. The form is used to select relevant pieces of information in the input articles. The user is then presented with a succession of extracted text fragments from the input articles. SumItBMT does not attempt to assemble these pieces of information into a single summary. In addition, the summaries are targeted to physicians only and are not adaptable to the comprehension level of non-physicians.

2.3 Text-to-text Generation and Text Simplification

Adapting a technical text to the user's level of expertise can be seen as a text-to-text generation problem. Text-to-text generation is an emergent field of NLP where an input text is transformed into a new text according to a set of constraints.

Examples of such constraints are length, style, and reading level. Despite recent attempts (Soricut and Marcu, 2005), there is no consensus yet on the framework or architecture of such systems as there is for concept-to-text generation systems (Reiter and Dale, 2000a).

There has been some work on sentence rewriting besides sentence compression. Mani, Gates, and Bloedorn (1999) propose to improve the quality of summaries by revising them through a set of rewriting rules. The rules are manually built, but they operate over a textual input parsed and annotated with coreference information. Rules can add or include text by the way of aggregation or insertion of additional information from the input article. This is in contrast to the earlier knowledge-rich approach of Robin (1994), who proposed rewriting rules to enrich the quality of generated texts by adding background information. Both these methods rely on manually built revision rules obtained through manual analysis of target texts.

While many agree that automatic text simplification is a valuable application, few researchers have actually tackled this problem. The only facet of text simplification which has been investigated is syntactic simplification. In their work, Chandrasekar and Bangalore (1997) propose a method to induce simplification rules automatically to edit syntactic parse trees. They rely on a manually built corpus of sentences paired with corresponding artificially manually simplified sentences. Carroll et al. (1999) describe a set of rules to simplify newspaper texts for aphasic readers using a set of manually defined syntactic and lexical rules. The types of simplifications obtained through such systems may seem disappointing from a human viewpoint. For instance, a sentence containing two clauses will be split into

two sentences, with one clause per sentence. However, not all text simplification systems are designed to help end users understand texts better. As a matter of fact, syntactic text simplification has been used mostly to provide an intermediate, simpler stage for NLP processing, such as parsing or content selection in a summarizer (Siddharthan, Nenkova, and McKeown, 2004). In our work, we are interested in a different type of simplification, not defined in terms of syntax or lexicon. We aim to transform a text technical in nature into a simpler version. The goal is for a given technical sentence to be edited to make its wording comprehensible to lay readers.

One area which promises to boost research in text-to-text generation is recent work in paraphrasing. A text-to-text generation system could rely on a collection of paraphrases to select the one that fits the constraints of the output text the best. This would depart from traditional generation systems where the lexicon, the dictionary mapping semantic concepts to alternative realizations, was manually built by domain experts (Reiter and Dale, 2000b). An automatically generated lexicon, or at least, an automatically augmented lexicon with paraphrases, would lower drastically the cost of designing a generation system. However, if there has been much interest in studying paraphrases (McKeown, 1979; Meteer and Shaked, 1988; Iordanskaja, Kittredge, and Polguere, 1991; Robin, 1994; Dras, 1999) and acquiring paraphrases automatically (Inui and Hermjakob, 2003), no such generation system has been implemented yet. There are two notable exceptions, however. Barzilay and Lee (2003) propose an unsupervised method for learning sentence-level paraphrases and their generation. In their framework, paraphrases are not classifiable into a syntax/lexical paradigm; multi-sequence alignment is applied on sets of

sentences that convey similar information, and the alignment is stored as a lattice. New sentences can be generated by following a specific path in the lattice. Similarly, Pang, Knight, and Marcu (2003) align sets of semantically equivalent translation sentences at the syntactic level into finite state automata. Generation of new sentences is achieved by traversing the FSA. In our framework, we do not have access to sets of semantically equivalent sentences, or any corpus of paraphrases. We instead rely on general resources, which we know to be comprehensible to lay readers.

Chapter 3

Content Selection

This chapter presents our strategy to identify which pieces of content to include in a summary. Traditionally, criteria for salience are determined in terms of the input documents only, but in our framework we rely not only on the input documents but also on a user's individual interests, as expressed in a user model. There are two main challenges we aim to solve at this stage of the summarization process: (1) what data structure is appropriate for representing and manipulating content given the summarizer's characteristics, and (2) what is an operative definition of relevance with respect to a set of user characteristics.

The contributions of this chapter are two-fold. First, information in the user model and the input articles is encoded in a representation that is between full-semantic analysis and pure extracted text. Semantic information enables us to manipulate different pieces of information, at this stage of the summarizer to compare them against each other and at the later stages of the summarizer to cluster them and to verbalize them back into fluent text. At the same time, the representation is still shallow enough to be obtained in a robust fashion automatically. Our

second contribution is that the content selection is sensitive to the user's interests, as modeled by the patient record. Rather than selecting relevant pieces of information at once, we adopt a two-step approach. The first step is concerned with selecting the type of information that we know should be in the summary (content extraction), and the second one filters the candidate information pieces to keep only information relevant to the user (content personalization). The filtering operates over our representation and follows simple and efficient personalization strategies. Breaking down the problem of finding relevant content into two conceptually independent subproblems (extraction and personalization) simplifies the task and yields a more robust strategy for content selection.

We first describe how salient information is represented internally, both for the information present in the input articles and the information present in the user model. We then go through our methods to extract content from the input articles and personalize it against the user model. After comparing our work to related research, we report on an intrinsic evaluation of the content selection module.

3.1 Content Representation

Based on the summarizer's goals and characteristics, we have defined the summary content as the results that (1) are conveyed in the input articles, and (2) are relevant to the user's interests as approximated by the electronic patient record of the patient under care. We, therefore, must come up with a representation that allows us to identify results in the input articles, as well as a representation of the patient record, such that it is easy to match the results from the input articles against the patient record.

3.1.1 Atomic Content Representation: Concepts

Definition

We base our representation, for both input article and patient record, on medical terms rather than all words. A Concept is principally defined as a medical term (e.g., "left ventricular ejection fraction," "heart attack," or "death"). In addition, some Concepts have an associated value ("low" in "low ejection fraction," or "35%" in "ejection fraction of 35%"), while some do not (e.g., "death"). A medical Concept can be verbalized in many ways: for instance, "heart attack" and "myocardial infarction" refer to the same Concept. We rely on the medical ontology of medical terms UMLS, or Unified Medical Language System (National Library of Medicine, 1995; McCray and Bodenreider, 2002), to gather a unique semantic identifier for each term. "Heart attack" and "myocardial infarction" are encoded in UMLS under the same entry or CUI (Concept Unique Identifier), *C0027051*, along with 39 other spelling/terms. Encoding the CUI into our data-structure enables us to have an atomic, semantically oriented representation of the medical terms used both in the input documents and the patient record.

In addition to the CUI, we encode a Concept's semantic type, as provided by the UMLS ontology (e.g., "heart attack" is coded under the type "Disease or Syndrome," while "left ventricular ejection fraction" is classified as a "Laboratory or Test Result"). This information provides an additional level of abstraction over the terms and their CUIs and proves useful when manipulating them, for instance when comparing Concepts against each other. We also store in our representation the way a given Concept is verbalized. This will come in handy at the generation stage of the summarizer; we can re-use the original strings to translate the summary

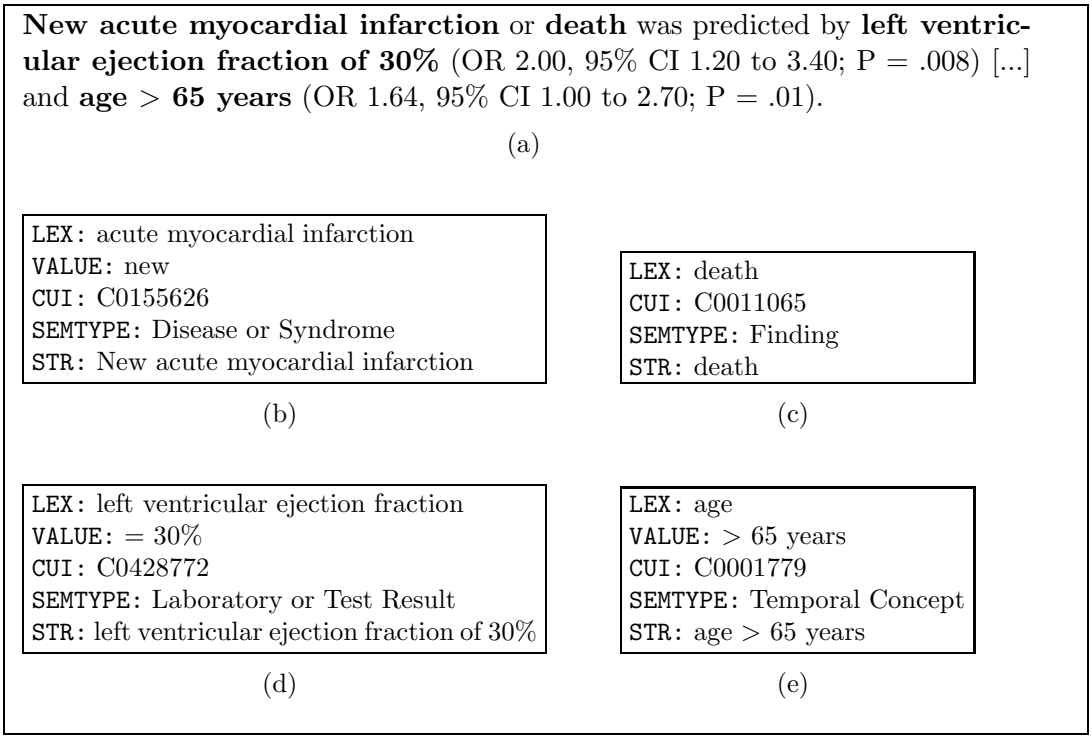


Figure 3.1: A sentence and some corresponding Concepts.

content into a fluent text.

A sentence and some Concepts it contains are shown in Figure 3.1. Below the sentence in Figure 3.1(a) is the representation of each highlighted term (b-d). In brief, we define the atomic representation, i.e., a Concept, with the following information as illustrated by the term 3.1(b): the original string used in the text to verbalize the Concept as a whole (`STR`), the verbalization used for the medical term (`LEX`), its associated value if any (`VALUE`), the term's identifier (`CUI`) and its semantic type (`SEMTYPE`), as extracted from the UMLS.

Concept Similarity

Several operations can be performed over Concepts. One is pairwise comparison to determine their degree of similarity. We approximate similarity as a three-way scale (no similarity, some similarity, full similarity). Given two Concepts, the algorithm first looks up the Concepts' CUIs. If the CUIs are different, then they are considered to have no similarity. It is a conservative decision that does not hold for every Concept pair. For instance, "atrial fibrillation" (CUI C0004238) is a type of "arrhythmia" (CUI C0003811), and one could consider these two Concepts to be somewhat similar. However, the two Concepts, in addition to having an *isa* relation in the UMLS, also share a *child*, a *sibling*, and a *narrower-than* relation. Conversely, the two Concepts "left ventricular ejection fraction" (CUI C0428772) and "ejection fraction" (CUI C0489482) are considered synonyms by most physicians, yet they are encoded in UMLS as having an "other relation than broader, narrower or synonym." Thus, the inter-concept relations provided by UMLS are too vague for a robust similarity measure to rely on.[1]

If the two Concepts have the same CUI, the following cases arise. If none have an associated value, they are considered to have full similarity. If one of the Concepts has an associated value, but not the other one, they are considered somewhat similar. Finally, if both Concepts have an associated value, the algorithm tries to match the values. In our framework, several types of values can be associated with medical terms: numerical with an exact number (e.g., in Figure 3.1(d)),

[1]In addition to the relations holding between individual Concepts, UMLS contains a complex semantic network, which links the different semantic types occurring in the ontology under 54 different relations. This, however, is not helpful for our purpose, since these high-level relations do not necessarily apply to all the instances of Concepts that have been assigned to those semantic types.

numerical with a range (e.g., in Figure 3.1(e)), or qualitative (e.g., in Figure 3.1(b)).

Given a medical term, matching values automatically is a difficult task. In order to compare two numerical values, whether an exact value or a range, one needs to know the typical value distribution for this term. For example, there is no way of knowing whether 1.2 and 1.5 are similar values for a term, unless we know that the values typically range from 1 to 100 (e.g., as they would for the cardiac test Total CK for instance) and not from 0 to 2 (e.g., as they would for a another cardiac test Troponin). As for qualitative values, we could not find any existing comprehensive list of terms along with ranges corresponding to qualitative judgments (such as, "systolic blood pressure under 90 mm Hg is low, between 90 and 120 mm Hg is normal, and above 120 mm Hg is high"). We tried to learn these relations automatically from a large corpus of texts, but our method allowed us to do so only for a limited set of terms, which is not sufficient to affect this similarity primitive in general (the amount of data constrained which terms we could look at). We, therefore, follow a conservative comparison method once again: two Concepts are considered fully similar if their values are easily matchable (same qualitative value or same numerical value), and somewhat similar otherwise.

3.1.2 The User Model as a List of Concepts

An electronic patient record typically contains many individual reports, collected during the patient's visit(s) to the hospital. For some patients, there can be up to several hundred reports. While some of the reports are in tabular format, and thus similar to a database entry (e.g., microbiology reports), many of the reports are textual (e.g., they may be the result of dictation, as is the case in operative

PHYSICAL EXAMINATION: On admission, a pleasant, pale appearing man, **BP** was 130/90, **heart rate** was 66 bpm. **JVD**, up/lying flat, **lungs** clear to **percussion** and **auscultation**. **Heart** exam was S1, S2 with a II/VI **holosystolic murmur** at the left upper sternal border. **Abdomen**: soft, no **hepatosplenomegaly**. **Extremities**: No **edema**. Neurologically intact.

(a)

Term	CUI	Sem. Type	Context	Value
PHYSICAL EXAMINATION	C0031809	DIGN	+	
BP	C0005823	PFUN	+	130 / 90
heart rate	C0018810	PFUN	+	66 bpm
JVD	C0240100	FIND	+	
lungs	C0024109	PART	+	
percussion	C0030987	DIGN	+	
auscultation	C0004339	DIGN	+	
Heart	C0018787	PART	+	
holosystolic murmur	C0232258	FIND	+	II / VI
Abdomen	C0000726	PART	+	
hepatosplenomegaly	C0019214	FIND	-	
Extremities	C0015385	PART	+	
edema	C0013604	DISS	-	

(b)

Figure 3.2: Extract from a patient discharge report and its corresponding representation.

reports) and thus require additional processing. We represent the information in the patient record/user model as a list of Concepts, along with contextual information to determine whether the Concepts appear in a positive or negative context. Figure 3.2 shows a paragraph from the discharge report of a patient, along with the corresponding list of Concepts. "Edema," for instance, is stored with a negative context to reflect the fact that the patient record says "No edema." We do not attempt to make any inferences on the information extracted from the patient record. For example, the blood pressure number reported in Figure 3.2 is one recorded at the time of physical examination. The patient report may very well have other

Multivariable analysis identified *diabetes, estrogen therapy (adjusted risk ratio 0.38, 95% confidence interval 0.19 to 0.79) and left ventricular ejection fraction < 40%* as independent correlates of **cardiovascular death** or **myocardial infarction** during follow-up.
Both *smoking status and insulin sensitivity* were independently related to both **insulin release and lipid levels**.
New acute myocardial infarction or death was predicted by *ST-segment depression (OR 2.00, 95% CI 1.20 to 3.40; P = .008), prior angina (OR 2.70, 95% CI 1.34 to 5.57; P = .001), and age > 65 years (OR 1.64, 95% CI 1.00 to 2.70; P = .01).*
Significant multivariable predictors of **later atrial fibrillation** included *advanced age, higher peak creatine kinase levels, worse Killip class and increased heart rate.*

Figure 3.3: Result sentence examples. The Parameters are in italics, the Findings are in bold, and the Relations are underlined.

mentions of blood pressure at different times (such as pre or post surgery). Determining the context for each piece of information and its representation is a research challenge that we do not address in this thesis. Thus, the user model will simply contain each mention of the blood pressure numbers, without attempting to place them in any global context.

3.1.3 The Articles as a List of Content Items

Concepts by themselves do not provide enough information to allow for identification of results in the input articles. Examples of sentences that convey results are given in Figure 3.3. We consider a sentence to be a *result sentence* if it reports *parameters* (such as disease, therapy, or patient characteristic) related to a *finding* (such as cure, disease, or other outcome). Since we are interested in summarizing result sentences, we encode as Content Items only the information conveyed in such

sentences.

Formally, a Content Item is represented as the triplet: (*Parameter(s), Relation, Finding(s)*). Parameters and Findings are Concepts, and the Relation is one of six relations manually identified through a corpus analysis: *association, prediction, risk, absence of association, absence of prediction,* and *absence of risk*. For instance, the first sentence in Figure 3.3 can be represented by a Content Item with three Parameters (diabetes, estrogen therapy, and left ventricular ejection fraction) in an association Relation with two Findings (death and myocardial infarction). Accordingly, the corresponding Content Item will be *(diabetes, estrogen therapy, left ventricular ejection fraction, association, death,myocardial infarction)*. The six types of Relations are not independent from one another; for instance, a *prediction* assumes an *association*. However, they reflect the language used by physicians when reporting results. Preserving this piece of information all the way to the generation step allows for the production of language that best matches the input text.

3.2 Content Extraction

3.2.1 Concept Tagging

Tagging the Concepts in a text is not a matter of simply looking up all the strings in a dictionary. The UMLS is a very noisy dictionary; while some entries would fit our definition of Concept, some would not. For instance, the phrase "Assess risk factors (smoking, obesity, stress, occupation)" is an entry in UMLS (coded under the CUI C0547253 and the semantic type "Therapeutic or Preventive Procedure"). To identify Concepts in a text, we make the assumption that a Concept is a noun phrase

or, at least, a head noun along with other words in a noun phrase. Accordingly, the text is first tagged with part-of-speech information. We use Brill's POS tagger for this task. We augmented the original lexicon provided by Brill with the UMLS lexicon to fit better the style of clinical studies. For instance, the word "univariate" was added from the UMLS lexicon (it is encoded there as an adjective). Next, a shallow chunker identifies noun phrases and matches them against the UMLS meta-thesaurus. If the entire noun phrase is not an entry in UMLS, the right-most portion of it gets matched, and so on. This way, the whole phrase "left ventricular ejection fraction," rather than simply "ejection fraction" gets identified as one term. Values are identified using a small set of regular expressions, such as "a symbol followed by a string of numbers and units that appears after a term." So for instance, the value "=30%" will be identified in the phrase "left ventricular ejection fraction of 30%."

When matching against UMLS, we look for an exact match in the meta-thesaurus. However, the meta-thesaurus contains many terms which we do not want to consider terms for the purposes of our application. For instance, "patient" is listed in UMLS, but we would rather process this word as a regular noun, not a medical term. To confirm this intuition, we asked six attending physicians and medical students to highlight the phrases they consider to be medical terms in three articles. Their selection shows that while they consider medical terms any phrase of the type "Body Part," "Medical Device or Procedure," "Disease Name," or "Treatment," they ignore phrases expressing localities, administrative tasks, etc. We, therefore, manually pruned terms in the UMLS meta-thesaurus based on their semantic types. In addition, because the terms are annotated with a hierarchy of

semantic types at a fine grained level – there are 135 different semantic types in UMLS – we manually grouped some related semantic types under a broad label, to simplify further processing. For instance, the semantic types "Rickettsia or Chlamydia," "Fungus," and "Alga" were grouped among others under the type "Organism." Appendix C shows the hierarchy of the 135 original semantic type and our modifications.

A given lexical item can link to several CUIs in UMLS (in the version of UMLS we used, there are 17,361 ambiguous lexical items). For instance the string "cold" can refer to "cold temperature" (CUI C0009264), "common cold" (CUI C0009443), "cold therapy" (CUI C0010412), "chronic obstructive airway disease" (CUI C0024117), "cold sensation" (CUI C0234192), and "Cold brand of chlorpheniramine-phenylpropanolamine" (CUI C0719425). Furthermore in some cases, the same CUI can be coded under different semantic types. Word sense disambiguation is a notably hard problem. Some ontologies help their users to some extent to disambiguate terms (WordNet, for instance, orders the different senses of a word based on their frequency, so that the most common sense is the first available). UMLS does not provide any mechanism to help disambiguate the terms or their semantic types. We do not attempt to disambiguate the CUI and semantic type correctly; instead, we assign the first CUI listed in UMLS, and similarly for the semantic type.

3.2.2 Content Item Tagging

Since clinical studies follow strict stylistic conventions, it is known in advance that the result sentences will be reported in the "Abstract" and the "Results" sections.

However, not all the sentences in these sections report results; out of an average of 20 sentences, approximately a third are actual results. To gather Content Items, we use information extraction techniques.

Sentences are first parsed using a shallow syntactic parser to identify noun and verb phrases. Because existing state-of-the-art parsers are trained on newspaper texts, their accuracy drops significantly when used for technical medical texts.[2] In clinical studies and more specifically result sentences, nested parenthetical phrases and numerical phrases appear often and cause the drop in accuracy. We customized the shallow parser CASS (Abney, 1996) to the style of our target sentences by providing our own manually written grammar. The resulting parser yields high-precision parses when it comes to result sentences (but not, as expected, for other types of sentences present in clinical studies). One of the helpful rules we introduced in our grammar concerned semantically empty words like "presence" in constructs such as "presence of atrial fibrillation," or "admission" in "heart rate at admission." Our CASS rule specifies that the Concept is the head for the noun phrase in all cases.

Once the sentences are parsed, templates are instantiated using a set of extraction patterns. While there has been work done on learning extraction patterns (Riloff, 1996; Riloff and Jones, 1999), it is not clear how to adapt these techniques easily to the present task. Such techniques commonly assume that there is a one-to-one mapping between the possible semantic tags of the lexicon and the slots of the templates to instantiate. In our case, this assumption does not hold. Given this difficulty, the learning of patterns in the medical domain is a challenging

[2]See Clegg and Shepherd (2005) for a study of parser's accuracies when applied on biomedical corpora.

task, which is not addressed in this thesis. We instead collected a set of hand-written patterns which rely on the syntactic information provided by CASS and lexical cues. Overall forty two patterns were gathered by analyzing a small corpus of clinical studies. Examples of patterns are shown in Figure 3.4. The first pattern, for instance, will match up with the first sentence from Figure 3.3. Because both the Parameters and the Findings can be verbalized by noun phrases with the same semantic types (any medical term), the patterns are global — all the slots in the template (Parameters, Findings, Relation, etc) are filled at the same time.

Figure 3.5 shows an example of an instantiated template, or Content Item. The following information is stored: a unique identifier (`ID`), the source article (`FILE No.`), the sentence number from within the article (`SENTENCE`), the type of Relation linking the Parameters and Findings (`RELATION`), whether the Parameters are independent in the statistical analysis reported (`ANALYSIS_TYPE`), a list of Findings (`FINDINGS`) and a list of Parameters (`PARAMETERS`). The Findings and Parameters are Concepts, as described earlier.

At the end of the content extraction stage, the extracted Content Items form the potential summary content representing the results presented in the input articles. If we were to produce a generic summary, we could now go on to the task of organizing these individual pieces of content into a fluent text. Our goal, however, being to produce a user-sensitive summary, we need first to determine which content is actually relevant to the user model and which is not and should, therefore, be discarded. The next section reports our strategy to filter the set of candidate Content Items.

analysis <VX HEAD='identified'>.*</VX> **(<NG>.*</NG>)** as <NP HEAD='predictors'>.*</NP> of *(<NG>.*</NG>)*
(<NG>.</NG>)* <VX HEAD='related'>.*</VX> to **(<NG>.*</NG>)**
(<NG>.*</NG>) <VX HEAD='predicted'>.*</VX> by *(<NG>.*</NG>)*
predictors of **(<NP>.*</NP>)** <VX HEAD='included'>.*</VX> *(<NG>.*</NG>)*

Figure 3.4: Extraction pattern examples. They are encoded as regular expressions. The elements are words (W), noun phrases (NP) and noun groups (NG). The Parameters are in italic and the Findings are in bold.

3.3 Content Personalization

The content extraction stage is responsible for identifying the type of information that we know should be in a summary (results, as opposed to other content present in clinical studies, such as background or methods of the study). It provides a shallow semantic representation of the results — Concepts are identified and tagged as Parameters or Findings, and, along with the Relations that link them, they form the different extracted Content Items. In addition, the user model also provides some shallow semantics, but about the patient record, in the form of a list of Concepts. Personalization takes advantage of the semantic representations provided by the Content Items and the list of Concepts in the user model. Our personalization strategy takes as input a Content Item and a list of Concepts and either returns a filtered version of the Content Item or discards the item entirely. The strategy relies on a function that matches the Concepts present in the Content Item against the Concepts present in the user model.

Besides making the summary content relevant to the user model, the person-

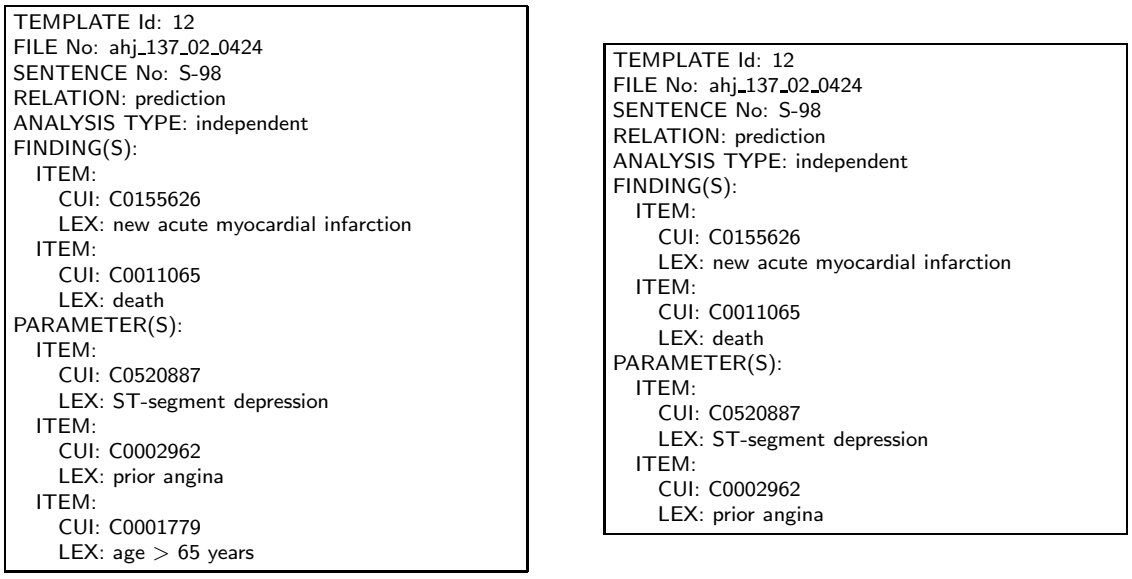


Figure 3.5: Content Item before and after content personalization. The Content Item was instantiated from the sentence "*New acute myocardial infarction or death was predicted by ST-segment depression (OR 2.00, 95% CI 1.20 to 3.40; P = .008), prior angina (OR 2.70, 95% CI 1.34 to 5.57; P = .001), and age > 65 years (OR 1.64, 95% CI 1.00 to 2.70; P = .01).*"

alization stage heightens the likelihood that the selected content will be coherent. Without personalization, the output would reflect results sentences derived from as assortment of independent input articles. When we personalize, we select only those Content Items pertaining to a single patient and his/her medical situation. Thus, we gain some assurance that, from a medical standpoint, the Content Items presented will tend to be related, so that it makes sense to convey them in a single output text.

3.3.1 Filtering Strategy

A Content Item is filtered by making recourse to the parameter(s) only, not the Finding(s). Consider the Content Item in Figure 3.5, which includes death as one of its Findings. From a medical standpoint, this Finding, which represents an outcome, will only become relevant to the patient if (s)he fits any of the Parameters that predict the given outcome. Thus, it is Parameters that determine whether or not a Content Item pertains to a given patient. Furthermore, some Parameters, standing alone, relate to given outcomes; other Parameters only do so in combination. For example, smoking can predict lung cancer, while smoking combined with pregnancy is associated with retardation of fetal growth. The dependence or independence of Parameters is identified when the Content Item is instantiated and is stored under the ANALYSIS_TYPE slot. In the case of independent Parameters, as in Figure 3.5, filtering simply keeps the Parameters that are relevant to the patient and throws away all others. In the case of dependent Parameters, the whole template is discarded unless *all* the Parameters are relevant to the patient. We next describe our algorithm to determine relevance of a Parameter or Concept with respect to a patient or user model.

In Figure 3.5, the three parameters ("ST-segment depression," "prior angina," and "age $>$ 65 years") are filtered against the user model. In our example, the patient is a 44 year-old female, hence the "age" Parameter gets filtered out. The term "angina" is mentioned in her patient record and, thus, carries on to the personalized template. Similarly for "ST-segment depression."

3.3.2 Concept Matching

The relevance of a Parameter to the user model is determined via a decision tree, and the pseudo code for the primitive is given in Algorithm 1. The tree encodes questions such as, "is the Concept mentioned in the user model?", "what is the semantic type of the Concept?", "is there a value associated with the Concept?", and "does the value of the Concept in the user model match the value of the Concept?"

Depending on the Concepts, several configurations can arise. If the Concept is mentioned in the user model but does not have any associated value, then it is considered matching. Examples of such Concepts are diseases (e.g., diabetes). If the Concept has a value and is mentioned in the patient record, then it matches only if the value of the Concept and the one mentioned in the record are similar (as defined by our Concept similarity primitive described in 3.1.1). Examples of such Concepts are laboratory results (e.g., "left ventricular ejection fraction of 30%"). Finally, some Concepts, which are not mentioned in the patient record, are nevertheless relevant to the patient. For instance, we consider any Concept with a semantic type of "Body Part" to match the user model, independently of whether or not it is mentioned. We refer to such semantic types as global.

3.4 Related Work

In the literature on text summarization, summary content is represented in many different ways, from the most shallow representation based on surface features of a text to a deep semantic representation obtained through understanding.

```
inputs : ParConcept: a concept from a template parameter
         UserModel: the user model
output: match/ noMatch

UMConcept ← lookup(ParConcept.CUI, UserModel );
if UMConcept then
    if (! ParConcept.value) then
        return match ; // the CUIs match and there is no value to compare
    if valuesMatch( ParConcept.value, UMConcept.value) then
        return match ; // the CUIs match as well as their associated values
else
    if isGlobalSemanticType( ParConcept.semantic_type) then
        return match ;
end
return noMatch ;
```

Algorithm 1: Concept matching primitive.

On one end of the spectrum, the systems that store a shallow internal representation of content typically function at the sentence level and are extractive summarizers. Representations include word vectors, semantic hyperlinks (Salton et al., 1997), lexical chains (Barzilay and Elhadad, 1998), or a vector of shallow features such as sentence length and position in text (Kupiec, Pedersen, and Chen, 1995). Such representations allow the system to identify the importance of a given sentence within a text, but no effort is made to process the sentence further. A similar representation to ours, but in a domain-independent setting, is presented in recent work on event-based summarization (Filatova and Hatzivassiloglou, 2004). An event is defined as any triplet (name entity, verb, name entity) occurring among several texts with sufficient frequency. Such events are similar to our Content Items, except that the slots are not linked to any semantic category such as Parameters, or Findings in our case. At present, this representation has been incorporated into

an extractive summarization framework. As in other systems that make use of shallow representation, the absence of semantic information and general knowledge of "what the event is about" limits the extent to which events can be used as-is for generative summarization.

A few summarizers aim to process the important sentences further, but without fully understanding the input text. Cut-and-paste summarization (Jing and McKeown, 2000) and summarization through sentence compression (Knight and Marcu, 2000) rely on lexical and syntactic information, in addition to cohesion cues, to determine which constituents of a sentence should remain in a summary and which ones can be dropped. MultiGen (Barzilay and McKeown, 2005) takes as input several sentences that convey similar information and fuses them. The sentences are represented in a dependency tree, as an approximation of a predicate-argument structure, and are aligned to identify an intersection sentence, which conveys what most input sentences convey but is also well-formed.

At the other end of the spectrum, summarizers that depend on a knowledge-rich representation use understanding techniques to extract and represent content. They typically operate over a specific domain, since much knowledge is required. The FRUMP system (DeJong, 1982), for instance, relies on scripts (Schank and Abelson, 1977). The input text is not interpreted; instead, FRUMP scans for the presence of specific keywords that activate script-like hand-coded structures to be instantiated.

Our approach to content extraction is similar to the ones of concept-based abstracting (Paice and Jones, 1993) and the SUMMONS system (Radev and McKeown, 1998). Concept-based abstracting (Paice and Jones, 1993) relies on informa-

tion extraction techniques for a domain-specific (crop husbandry) single-document summarizer. The summarizer fills out the slots of a given template (e.g., `AGENT`, `SPECIES`, `CULTIVAR`, `PEST`), one template per document. Template-based generation is used to output an indicative summary (e.g., ``` ``This paper studies the effect of [AGENT] on [HLP] of [SPECIES]'' ```). In our framework, however, several templates are instantiated (one per Content Item), without knowing in advance how many there will be. Template-based generation is not a viable option for us, as additional processing – organizing the Content Items into a coherent output – needs to be carried out. SUMMONS (Radev and McKeown, 1998) also operates for a specific domain (terrorist attacks). The input are pre-existing MUC-style templates and focuses on organizing the different templates into a summary. Because the templates contain a great deal of information, SUMMONS is able to identify updates, repetitions, and contradictions precisely. Our representation in the form of Content Items does not contain as much semantic information. We opted for a coarser-grained approach to content extraction in order to extract Content Items in a robust fashion. Nevertheless, our representation is detailed enough to allow for the filtering of Content Items against the user model and for the identification of some inter-textual phenomena (repetitions and contradictions), as will be shown in the next chapter.

3.5 Evaluation

In this section we report on the intrinsic evaluations carried out for the different steps of the content selection module.

3.5.1 Extraction

Concept Tagging

Using an augmented lexicon and filtering the UMLS terms yields good results for the Concept tagging task. We collected a gold standard of 20 articles and asked a physician to highlight all the medical terms she identified, along with their possible associated values. Overall, the gold standard contained 661 different terms. Over the same corpus, our tool achieves 83.6% precision and 95.5% recall. The precision was hurt mainly by our noun chunking strategy. For instance, the phrase "beta- and channel blockers" was tagged to contain only one term "channel blockers." Ideally, the term "beta blocker" should have been identified. Another factor was the high number of entries in UMLS that are not terms according to our working definition. For instance, "strategy" is recognized as a noun phrase and is an entry in UMLS. While, as noted above in Section 3.2.1, we did filter out much of the terms by ignoring the entries under some semantic types (e.g., "Professional or Occupational Group"), this was only a rough filtering.

Where terms were missed, it was largely a result of the coverage of UMLS itself. For instance, "New York Heart Associations class" should be recognized as a term, but UMLS only has "New York Heart Associations class III/IV" listed as a term, and not the more general one. Other terms were missing due to part-of-speech tagging mistakes. Although this tool was not formally tested on different medical genres, it has been used for other applications with success, such as medical textbook summarization and information extraction. The identification of values associated with terms achieved 85.4% accuracy.

Shallow Parsing

While the verb phrases in our domain are similar to the ones in more typical domains, noun phrases are harder to identify accurately. The customized version of CASS yields better results in identifying noun phrases than off-the-shelf parsers. When evaluated on 70 sentences, which contained 501 noun phrases, the shallow parser identified noun phrases with 82.4% recall and 73.6% precision. The identification of the noun phrase heads achieved 97.1%.

Content Item Tagging

To test the performance of the Content Item tagging, we asked a physician to read 40 clinical studies from seven different journals and annotate any sentence reporting a result with Parameters, Relation, Finding. Our extraction tool achieved 89% precision and 65% recall when run on the same studies. An instantiated template was considered accurate if all its slots got instantiated correctly. The low recall is to be expected given that the extraction patterns were manually written.

3.5.2 Personalization

We asked the medical expert part of the PERSIVAL project to select the result sentences pertaining to a patient record with respect to a medical scenario for three different cases.[3] Each case contains seven to eight articles. When comparing our final selection at the sentence level against the expert's selection, we achieve on average 71.2% precision and 30.6% recall. Table 3.1 shows the precision and recall counts for each scenario.

[3]The scenarios are described in more detail in Chapter 6.

Scenario	Precision	Recall
Scenario 1	29.4%	17.9%
Scenario 2	100.0%	25.0%
Scenario 3	89.3%	38.5%

Table 3.1: Precision and recall of the personalization strategy for three evaluation scenarios.

Since personalization relies on the tagging of Content Items, it is affected by the tagging performance. The low recall counts of the tagging stage limit the recall of personalization. The personalization strategy yields high precision in two out of the three scenarios, the second and the third. Upon speaking with our medical expert to analyze the results of the first scenario, we realized that, for this scenario in particular, the expert had relied heavily on his general medical knowledge. Several articles dealt with the drug quinidine to treat atrial fibrillation, a disease mentioned in the patient record. The expert did not select any of the Findings related to this drug, because he knew from his experience that the drug would actually be dangerous if administered to the patient in question. The summarizer, having no access to such a knowledge base, selected these Findings. This scenario demonstrates that personalization, in theory, cannot be done in a vacuum, without relying on a medical inference engine. Nevertheless, in less unusual cases, these results show that personalization is feasible, even with a minimal amount of knowledge.

Chapter 4

Content Organization

This chapter describes the organization stage of the summarizer. Once the different pieces of information have been selected from the input articles and filtered by the user model, the organization stage is responsible for ordering them. Organizing information is a task particularly important in multi-document summarization, whether extractive or generative. Applying the simple strategy employed by most single-document summarizers – present each piece of information in the order it appeared in the input article – is not possible as our information comes from different articles, and no single document can provide a complete ordering. This is especially so given the fact that the order of two pieces of information can differ from one input article to another.

We contribute to research on content organization in three ways: (1) We conducted a study to verify the impact of ordering on the overall quality of summaries. Our analysis shows that ordering significantly affects the reader's comprehension of a summary. (2) We established a corpus-based methodology for studying ordering. Supervised learning methods are not easily applicable to our problem in part be-

cause of lack of training data. Given that there are multiple possible orderings, a corpus providing one ordering for each summary does not allow us to differentiate between pieces of information which happen to be together and pieces of information which have to be together. We developed a corpus of several data sets, each of which contains multiple acceptable orderings of a single text. Such a corpus is expensive to construct and, therefore, does not provide enough data for a pure statistical approach. Instead, we complemented our study with an automatic analysis, which identifies commonalities across orderings. (3) We incorporated the results of our ordering study into a novel ordering strategy for multi-document summarization. Instead of ordering each item of information individually, a better choice is to first figure out which items should be presented together and then order each group of items. The strategy is easily adaptable to new domains. It is currently integrated in two summarizers: TAS and MultiGen (Barzilay and McKeown, 2005), which operates over news articles.

This chapter is organized as follows: We first describe our study to verify the impact of ordering on the overall quality of a summary. We then explain our methodology to derive an ordering strategy for summarization: the collection and analysis of a corpus of multiple orderings. We follow by presenting our ordering strategy as implemented in TAS. Finally we show evaluation results and discuss related work.

4.1 Impact of Ordering on the Overall Quality of a Summary

Even though the problem of ordering information for multi-document summarization has received relatively little attention, we hypothesize that good ordering is crucial to produce summaries of quality. It is clear that ordering cannot improve the output of earlier stages of a summarizer, among them content selection; however, finding an acceptable ordering can enhance user comprehension of the summary and, therefore, its overall quality. To verify our hypothesis, we performed a study to measure the impact of ordering on the user's comprehension of summaries.[1]

Because it is difficult to obtain existing machine-generated texts in the medical domain, we took advantage of existing machine-generated summaries in the news genre. In addition, since we wanted to test users' comprehension, we preferred to select texts in a general domain, and not in a technical domain, like the one in which our summarizer TAS operates.

We began by selecting ten machine-generated summaries from a system participating in the 2001 Document Understanding Conference (DUC). As part of the DUC evaluation, each summary was graded by human judges according to different criteria, among them how well the information contained in the summary is ordered. To actually identify a possible impact of ordering on comprehension, we selected only summaries where humans judged the ordering as poor. All ten summaries were produced by the DEMS system (Difference Engine for Multidocument Summarization) (McKeown et al., 2002), a good choice of summarizer for this study

[1]The work presented in this section and the following one is based on the JAIR paper (Barzilay, Elhadad, and McKeown, 2002).

Summary set	Original	Reordered
d13	Incomprehensible	Incomprehensible
d19	Somewhat comprehensible	Comprehensible
d24	Incomprehensible	Comprehensible
d31	Somewhat comprehensible	Comprehensible
d32	Incomprehensible	Somewhat comprehensible
d39	Incomprehensible	Incomprehensible
d45	Incomprehensible	Incomprehensible
d50	Incomprehensible	Comprehensible
d54	Incomprehensible	Somewhat comprehensible
d56	Comprehensible	Comprehensible

Figure 4.1: Impact of ordering on the user comprehension of summaries.

because it does not have any explicit ordering strategy, and thus provides us with appropriate data.

For each summary, we manually reordered the sentences generated by the summarizer, using the input articles as a reference. When manually reordering the sentences, we did not change the content; all the sentences in the reordered summaries were the same ones as in the originally produced summaries. This process yielded ten additional reordered summaries, and thus, overall our collection contains twenty summaries. Two subjects participated in this experiment. Each summary was read by one participant without having access to the input articles. We distributed the summaries between the subjects so that neither of them read both an original summary and its reordering. They were asked to grade how well the summary could be understood, using the ratings "Incomprehensible," "Somewhat comprehensible," or "Comprehensible."

The results are shown in Figure 4.1.[2] Seven original summaries were con-

[2]The set names are the ones used in the DUC evaluation.

sidered incomprehensible by their judge, two were somewhat comprehensible, and only one original summary was fully comprehensible. The reordered summaries obtained better grades overall: five summaries were fully comprehensible, two were somewhat comprehensible, while three remained incomprehensible. The improvement was statistically significant (p=0.07) using the Fisher exact test on the data (the Incomprehensible and Somewhat Comprehensible summaries were conflated into one category to obtain a 2x2 table).

It is striking that in some summaries with low grades, poor ordering is the likely culprit. For instance, readers can easily identify that the order of the two following sentences is an unsuitable choice and could be misleading. *"Miss Taylor's health problems started with a fall from a horse when she was 13 and filming the movie National Velvet. The recovery of Elizabeth Taylor, near death two weeks ago with viral pneumonia, was complicated by a yeast infection, her doctors said Friday."* But in other cases, information in a summary is poorly ordered, and readers cannot make sense of the text sufficiently to realize that a better ordering would increase their comprehension. Rather, the subjects, in interviews, tended to blame the content selection instead of ordering, even where the content was not the issue. Thus, ordering is not an isolated phenomenon; it can affect the quality of a summary as a whole.

4.2 Ordering by Blocks

In this section, we give the rationale for our ordering strategy, and explain our methodology to discover important features in ordering: the collection of a corpus of multiple orderings and its semi-automatic analysis.

4.2.1 Collecting a Corpus of Multiple Orderings

Sentences in a text can be ordered in a number of ways, and the text as a whole will still convey the same meaning. But the majority of possible orders are likely to be unacceptable because they break conventions of information presentation. One way to identify these conventions is to find commonalities among different acceptable orderings of the same information. Extracting regularities in several acceptable orderings can help us specify ordering constraints for a given input type. While there has been a recent effort to collect human-produced summaries of multiple documents, such collections do not fit our needs since we want to analyze *multiple orderings of the same summary content*. We created our own collection of multiple orderings produced by different humans. Using this collection, we studied common ordering choices and mapped them to strategies for ordering.

As we explained above in conjunction with the study of the impact of ordering on summary quality, it is difficult to obtain data in the medical domain. Recruiting medically-trained subjects to read and order medical texts is expensive. Consequently we collected data in the general news domain. In the following section, we describe how we carry over our findings into an ordering method for summaries of medical findings.

We collected ten sets of articles. Each set consisted of two to three news articles reporting the same event. For each set, we manually selected the "intersection sentences," that is, the sentences conveying similar information in at least two input articles. This manual selection simulates the content selection strategy used in the state-of-the-art summarizer MultiGen (Barzilay and McKeown, 2005) and ensures that ideal data was provided to the subjects. On average, each set con-

Participant 1	D B G I H F C J A E
Participant 2	D G B I C F A J E H
Participant 3	D B I G F J A E H C
Participant 4	D C F G I B J A H E
Participant 5	D G B I H F J A C E
Participant 6	D G I B F C E H J A
Participant 7	D B G I F C H E J A
Participant 8	D B C F G I E H A J
Participant 9	D G I B E H F A J C
Participant 10	D B G I C F A J E H

Figure 4.2: Multiple orderings for one set in our collection. A, B, ... J stand for sentences. Underlined are automatically identified blocks.

tained 8.8 intersection sentences. The sentences were cleaned of explicit references (for instance, occurrences of "the President" were resolved to "President Clinton") and connectives, so that participants would not use them as clues for ordering. Ten subjects participated in the experiment, and they each built one ordering per set of intersection sentences. Each subject was presented with the intersection sentences in a random order and was asked to order them so that they form a readable text. Overall, we obtained 100 orderings, ten orderings per set. Figure 4.2 shows the ten orderings collected for one specific set. The full collection of multiple orderings, along with the instructions given to subjects and the interface used by the subjects, is available at `http://www.cs.columbia.edu/~noemie/ordering/`.

4.2.2 Analysis

We first observe that a surprisingly large portion of the orderings are different. Out of the ten sets, only two sets had some identical orderings (in one set, one pair of orderings were identical while in the other set, two pairs of orderings were

identical). In other words, there are many acceptable orderings given one set of sentences. This confirms the intuition that we do not need to look for a single ideal total ordering but rather construct an acceptable one.

Looking at these various orderings, one might also conclude that any ordering would do just as well as any other. This is not the case. For a text with n sentences, there are $n!$ possible orderings, but only a small fraction of those are actually valid orderings. One way to validate this claim would be to enumerate all the possible orderings of a single text and evaluate each one of them. This would be doable for very small texts (a text of 5 sentences has 120 possible orderings), but not for texts of a reasonable size. A more feasible way to validate our claim is to get multiple orderings of the same text from a large number of subjects. We asked additional subjects to order one text of eight sentences. There are 40,320 possible orderings for these sentences. While 50 subjects participated, we only obtained 21 unique orderings, showing that the number of acceptable orderings does not grow as fast as the number of participants. We can conclude that only a small fraction of all possible orderings of the information in a text contains orderings that render a readable text.

If most orderings produced for a single summary are different, they still exhibit commonalities. In particular, some sentences always appear together. They do not appear in the same order from one ordering to another, but they share an adjacency relation. We will refer to these as "blocks." For each set, we identify blocks by clustering sentences across orderings. We use as a distance metric between two sentences, the average number of sentences that separate them over all orderings. In Figure 4.2, for instance, the distance between sentences D and G is

2. The blocks identified by clustering are sentences B, D, G and I, sentences A and J, sentences C and F, and sentences E and H.

When reading the sentences in each block, we observed that all the blocks in the experiment correspond to clusters of topically related sentences. These blocks form units of text dealing with the same subject and exhibit cohesive properties (even if they come from different input articles). In other words, all valid orderings contain blocks of topically related sentences. The notion of grouping topically related sentences is known as cohesion. As defined by Hasan (1984), cohesion is a device for "sticking" together different parts of a text. Studies show that the level of cohesion has a direct impact on reading comprehension (Halliday and Hasan, 1976). Our experiments confirm this finding: good orderings are cohesive; this is what makes the summary readable. Incorporating the cohesion constraint into our ordering strategy by opportunistically grouping sentences together would be beneficial. We describe next how we integrate cohesion into our ordering strategy.

4.3 Methods

4.3.1 From News Articles to Medical Texts

The analysis of our corpus of multiple orderings showed that, while humans order information differently, they tend to agree on which pieces of information should be presented together. In addition, blocks of information are formed based on their topical similarity. The question of how to order the blocks of cohesive information, as well as how to order the information inside each block, is left to investigate.

In news summarization, the ordering strategy adopted by most summarizers

to date is to order each piece of information based on the publication date of the corresponding article in which it is mentioned (or the earliest publication date, if the information is mentioned in different articles) (Radev, Jing, and Budzikowska, 2000; Lin and Hovy, 2002). The underlying intuition is that events reported in news are time-dependent, and a chronological rendering would do justice to the story being summarized. We developed a new ordering strategy for news summarization by applying a chronological ordering, not on each item individually, but on each previously identified block of related items. It was integrated in the MultiGen summarizer in the following way.

In a preprocessing stage, we segment each input article based on word distribution and coreference analysis (Kan, Klavans, and McKeown, 1998), so that given two sentences within the same text, we can determine if they are topically related. Assume the themes A and B exist, where A contains the similar sentences ($A1...An$), and B contains the similar sentences ($B1...Bm$).[3] We denote $\#AB$ to be the number of pairs of sentences $(Ai; Bj)$ which appear in the same text, and $\#AB^+$ to be the number of sentence pairs which appear in the same text and are in the same segment.

In a first stage, for each pair of themes A and B, we compute the ratio $\#AB^+/\#AB$ to measure the relatedness of two sentences. This measure takes into account both positive and negative evidence. If most of the sentences in A and B that appear together in the same texts are also in the same segments, it means that A and B are highly topically related. In this case, the ratio is close to 1. On the other hand, if among the texts containing sentences from A and B,

[3]A Theme in MultiGen is a set of sentences from different input articles that contain repeated information. It does not necessarily include sentences from every input article.

only a few pairs are in the same segments, then A and B are not topically related. Accordingly, the ratio is close to 0. A and B are considered related if this ratio is higher than a predetermined threshold. In our experiments on a small development set, we set it to 0.6. This strategy defines pairwise relations between themes. A transitive closure of this relation builds groups of related themes and, as a result, ensures that themes that do not appear together in any article but which are both related to a third theme will still be linked. This creates an even higher degree of relatedness among themes. Because we use a threshold to establish pairwise relations, the transitive closure does not produce elongated chains that could link together unrelated themes. At the end of this stage, topically related themes are grouped into blocks.

In a second stage, we assign a time stamp to each block of related themes using the earliest time stamp of the themes it contains. Blocks can now be ordered chronologically according to their time stamp. Finally, the themes inside each block are recursively ordered according to their own time stamp to form a complete order of the summary content.

If it makes sense to order pieces of information chronologically in the news domain, it does not when summarizing findings of clinical studies. Scientific findings are time-independent, and scientific articles are not narrative. While there certainly is an argumentative structure present in scientific articles, it is not clear whether there is an explicit underlying structure to the order in which findings are presented within scientific articles and, if so, how to capture it. As such, for TAS, we apply the same basis of our ordering strategy: grouping topically related information, whether themes in the news domain or Content Items in the medical domain, into

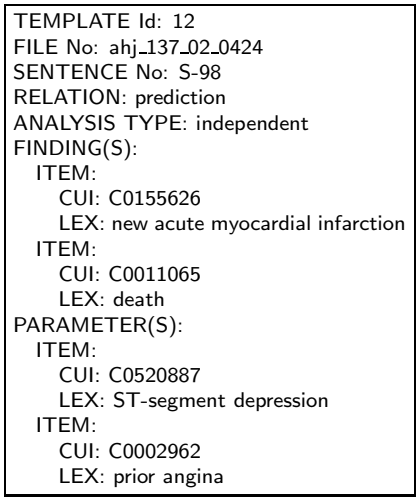


Figure 4.3: A Content Item before atomization. The Content Item was instantiated and personalized from the sentence "*New acute myocardial infarction or death was predicted by ST-segment depression (OR 2.00, 95% CI 1.20 to 3.40; P = .008), prior angina (OR 2.70, 95% CI 1.34 to 5.57; P = .001), and age > 65 years (OR 1.64, 95% CI 1.00 to 2.70; P = .01).*"

blocks; ordering at the block level; and ordering the information in each block independently of the other blocks. The remainder of this section describes the ordering component in TAS.

4.3.2 Finding Blocks of Content Items

While the different Content Items come from different input articles, they all pertain to the user model. In addition, the articles were selected (at the search stage of the PERSIVAL architecture) as relevant to the input question. Therefore, it is safe to assume that the Content Items are not a random set, and it is possible to obtain a somewhat cohesive structure out of them.

The Content Items instantiated at the content selection stage are first broken down, or atomized, into individual Content Items. For instance, the Con-

tent Item of Figure 4.3 can be broken down into the four Content Items of Figure 4.4: *(C0520887 [ST-segment depression], prediction, C0155626 [acute myocardial infarction])*, *(C0520887 [ST-segment depression], prediction, C0011065 [death])*, *(C0002962 [angina], prediction, C0155626 [acute myocardial infarction])*, and *(C0002962 [angina], prediction, C0011065 [death])*. From a medical standpoint, it is meaningful to atomize a Content Item only if the relationship between the different Parameters is independent, that is each Parameter is in an independent relation with the Findings. The slot *ANALYSIS_TYPE* in the template allows us to decide whether to atomize or not a specific Content Item. From now on, we only present the ordering strategy for the Content Items that can be atomized. For the others, we do not want to combine them with any other Content Item, as it could change its meaning. Thus, they end up as their own block.

To identify the blocks of related Content Items, the individual Content Items are clustered using hierarchical complete-link clustering. The similarity function between two Content Items is computed as the sum of the value of several features. Features are based on the Parameters' similarity (as defined by our Concept similarity primitive), the Findings' similarity, and how similar the Relations are.[4] In the case of Content Items with an "association" Relation, since it is bidirectional, the Findings and Parameters are interchangeable; the similarity function takes this fact into account.

Because the similarity function assigns higher weights to two Content Items with the same Parameters/Findings and Relations, repetitions are more likely to be grouped together. Similarly, contradictions are more likely to be clustered. A

[4]Each possible Relation pair was manually assigned a weight. For instance, (prediction, prediction) is weighted higher than (prediction, association).

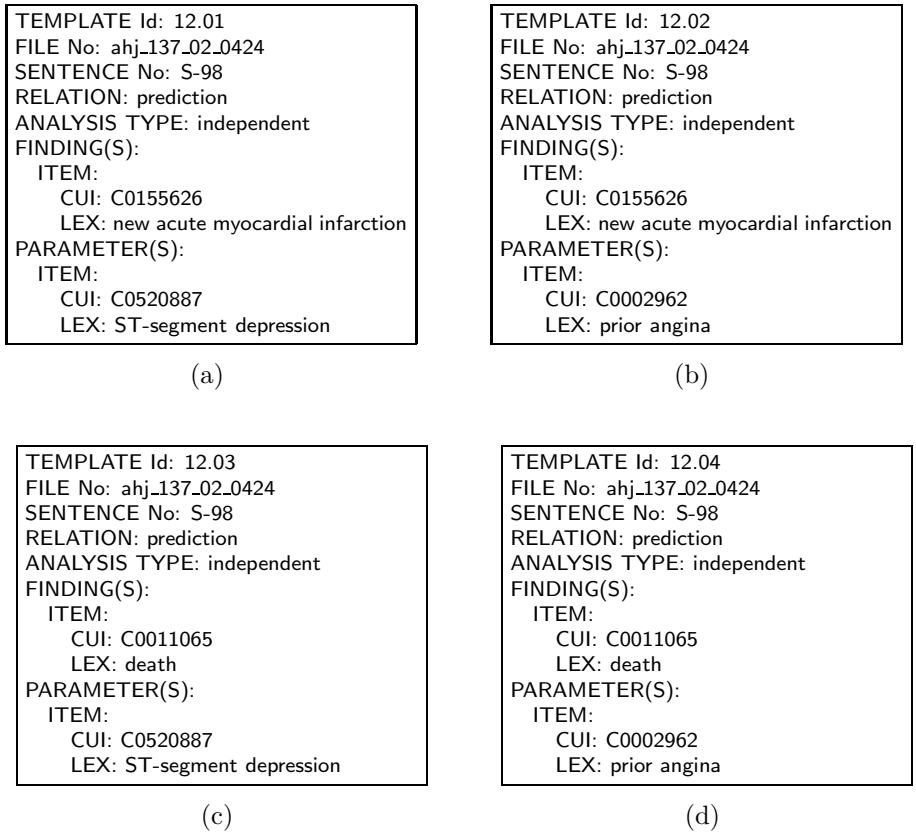


Figure 4.4: A Content Item after atomization.

contradiction is defined here as two Content Items that share the same Parameters and Findings but have contradictory Relations (e.g., "association" and "not association").

It is interesting to compare directly our criteria to recognize inter-textual phenomena, like repetitions and contradictions, against the ones recognized by the SUMMONS system (Radev, 1999). SUMMONS analyzes its input data to identify several phenomena: change of perspective, agreement between sources, contradictions, addition of information, refinement, and generalization. From a representation standpoint, TAS and SUMMONS organization modules differ in their input: The granularity of the SUMMONS templates, which enables the identification of

these phenomena, is much finer-grained than the templates used in TAS. More importantly, it is a design choice on the part of TAS to recognize such phenomena with caution. While in the news genre accuracy of information is important, it is critical in the medical domain not to alter the meaning in the input data. When it comes to contradictions, for instance, our definition is arguably conservative. It would be possible to consider a less conservative definition of a contradiction given the available data structure. For instance, one could define a contradiction as any two Content Items that have a contradictory relation, share the same Parameters, and contain Findings that are related semantically: *(chest pain, predict, atrial fibrillation)* and *(chest pain, not predict, arrhythmia)*. We choose to avoid making any complex inference. We present only the obvious contradictions as such and let the physician decide on the less obvious ones, instead of possibly presenting misleading information on the basis of erroneous medical inferences.

Clustering the Content Items into blocks achieves two purposes: (1) it allows for the easy identification of duplicate Content Items (that is, repetitions either across or inside articles) and of contradictory Content Items, and (2) it dynamically groups together the Content Items that are semantically related to each other. In the overall generation process of the summarizer, this step is equivalent to a dynamic paragraph planning, where each cluster or block represents a paragraph.

4.3.3 Ordering the Blocks

To determine an overall ordering of the summary content, we assign a priority to each block of Content Items. The block with the highest priority will be presented first to the user, the second highest, second, and so on.

During the initial user studies, physicians mentioned the two following features as important: facts that answer directly the initial question should be displayed at the beginning of the summary, and repetitions and contradictions should be presented early in the summary. In addition, a set of results is likely to be more important when it reflects input from multiple articles or at least from multiple analyses within the same article. Accordingly, each block of Content Items is assigned a priority based on the following counts: the number of Content Items that contain the terms asked about in the input user query, the number of repetitions present in the block, the number of contradictions present in the block, the number of different articles represented by the Content Items in the block, the number of distinct sentences represented by the Content Items in the block.

4.3.4 Ordering inside the Blocks

While the grouping of Content Items into blocks is equivalent to document planning, this next step is concerned with two goals: the identification of which Content Items should be presented together in the same sentence, and the order of the sentences inside the blocks. Our algorithm achieves both at once. Since we want the summary to be concise, the algorithm tries to combine as many Content Items as possible in each sentence through aggregation. Sentences are ordered according to how many Content Items they convey. At this point, we focus on conceptual sentence planning, as determined by aggregation. Sentence planning of Content Items from a pure generation standpoint is described in the next chapter.

In TAS, a subset of Content Items can be grouped into the same sentence for three reasons: (1) they are repetitions of each other, (2) they contradict each

other, or (3) they can be conceptually aggregated.

Repetitions

Following our operative definition of repetition, two (or more) Content Items are considered to repeat each other if they have the same Parameters and Findings and either share the same Relation, or one Relation can be subsumed by the other. For instance, the two Content Items in Figure 4.5 are repetitive because they both convey the fact that there is an association between "coronary artery disease" and "recurrent atrial fibrillation." However, because the first Content Item gives additional information, namely that the relation is a prediction, the two Content Items are merged into a prediction relation.

A merged Content Item is equivalent to a list of Content Items. Each Concept, whether a Parameter or Finding is kept, along with the articles and sentences they were extracted from. For instance, in Figure 4.5, the two verbalizations of the two Concepts "atrial fibrillation" and "AF" are stored, along with all the information contained in each Concept. It is important that the merging preserves the original Concepts and Content Items, as the information they contain will come in handy, as we shall see, during the sentence planning and the generation stages.

Contradictions

Like repetitions, contradictions are identified based on the operative definition of contradiction in the block ordering step: two Content Items that share the same Parameters and Findings but have contradictory Relations. Figure 4.6 gives an example of two contradictory Content Items. The first one conveys the fact that the

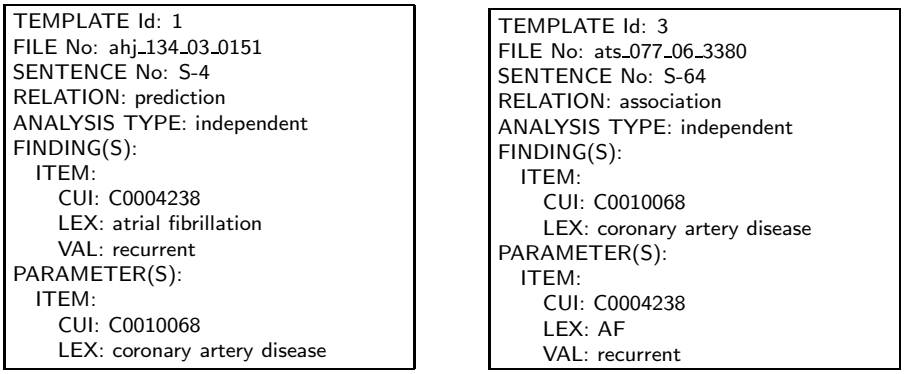


Figure 4.5: Two repetitive Content Items.

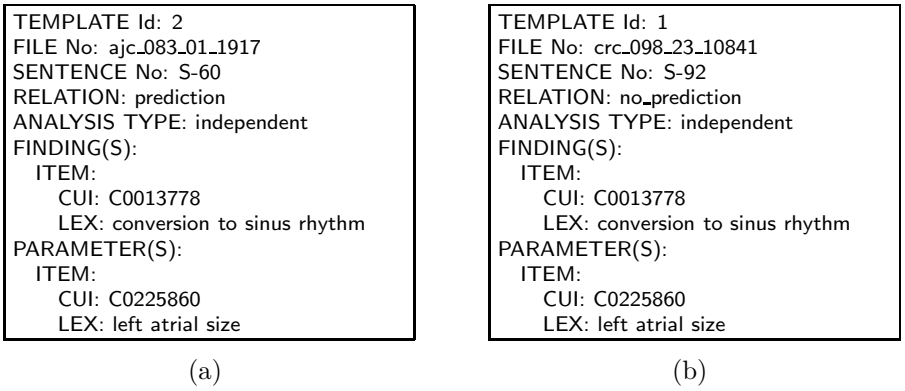


Figure 4.6: Two contradictory Content Items.

"size of the left atrium" predicts "conversion to sinus rhythm," that is, a normal heartbeat (as opposed to atrial fibrillation, for instance, which is a type of arrhythmia). For the second Content Item, in contrast, the authors of the extracted sentence found no such prediction relation between the two.

Aggregation

Two Content Items can be aggregated if they share the same Relation and a common Concept (either the Parameter or the Finding). For instance the Content

Items *(p1, prediction, f1)* and *(p1, prediction, f2)* can be grouped in the same sentence. They then can then be realized as "p1 predicts f1 and f2." Other examples of Content Items which can be aggregated are *(p1, prediction, f1)* and *(p2, prediction, f1)*; *(p1, association, f1)* and *(p2, association, f1)*. In the case of the association Relations, because there is no distinction between Parameters and Findings, the order of the two Concepts does not matter for aggregation purposes. Thus, *(p1, association, f1)* and *(f1, association, p2)* can also be aggregated.

As in the case of repetitions, the common Concept, repeated in every aggregated Content Items, is merged. Thus, all the verbalizations of the Concept are stored, as well as meta-information about their source (article and sentence numbers).

Ordering through conceptual sentence planning

The algorithm to group and order the Content Items in a block into sentences is given in pseudo-code in Algorithm 2. First, the repetitive Content Items are merged. Since we want to avoid duplication of information, it is important to merge repetitive Content Items before trying to do the sentence planning. Consider the three following Content Items in Figure 4.7(a): if we first combine them through aggregation, as in Figure 4.7(b), the first two Items will be realized first as "p1 is associated with f1 and f2," and then the third Content Item "p1 predicts f1" will be generated. These two sentences, however, are redundant. A more concise output is obtained by merging repetitions first (the first and third Content Items), and then looking for possible aggregation (in this case none), as in Figure 4.7(c).

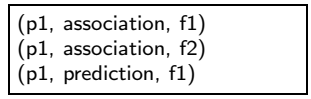
Once repetitions are identified and merged, the actual grouping of Content

```
input : contentItems: a block of Content Items
contentItems ← mergeRepetitions(contentItems );
contradictionPairs ← findContradictions(contentItems );
foreach pair ∈ contradictionPairs do generateSentence(pair );
removeContentItems(contentItems, contradictionPairs );
while contentItems ≠ ∅ do
    aggregateItems ← findLargetsAggregation(contentItems );
    generateSentence(aggregateItems );
    removeContentItems(contentItems, aggregateItems );
end
```

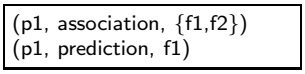
Algorithm 2: Conceptual sentence planning.

Items into sentences starts. Contradictions are identified and each pair is sent to the next generation module to be realized as one sentence. Thus, if there are any contradictions in the paragraph, they will be presented first to the user.

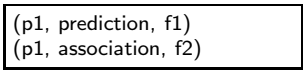
Next, the remaining Content Items are neither repetitions nor contradictions, yet are semantically related (since they are part of the same block). The grouping into sentences is driven by the assumption that the more Content Items are conveyed in one sentence, the more informative the sentence is, and the earlier it should be presented to the reader. Thus, our search is driven by the question "what is the largest number of Content Items that can be realized in the next sentence?" At each iteration, there could be several ways to aggregate the Content Items. Choosing the combination that aggregates as many Content Items as possible is a good approximation for achieving conciseness in the summary, and ordering the sentences inside a paragraph by how informative they are. Accordingly, the function `findLargestAggregation` looks for the largest subset of Content Items remaining in the block, that can be aggregated together according to a common predicate (either Parameter/Relation or Relation/Finding). The size of the subset



(a) Content of a block of Content Items.



(b) Aggregate first, merge second.



(c) Merge first, aggregate second.

Figure 4.7: Effect of aggregation and merging.

takes into account the fact that some items might be merged repetitions. Thus, in Figure 4.7(c), the merged Content Item *(p1, prediction, f1)* has in fact a count of two, compared to the single Content Items *(p1, association, f2)*, and, therefore will be generated first as a sentence. The algorithm guarantees that at every step of the iteration through the block the highest number of Content Items will be combined into one sentence, although it does not guarantees an optimal solution overall.

At this stage in the summarizer's architecture, the summary content has been selected, and the summary structure is determined, both at the paragraph and at the sentence level. The organization is such that redundancy is eliminated, contradictions are identified and grouped together, and the Content Items that can be combined together through aggregation are grouped and ready to be generated. Ordering by block, paired with our conceptual sentence planning prepares the ground for generating a coherent and concise summary.

4.4 Evaluation

Our ordering strategy was implemented in two summarizers: MultiGen and TAS, each in a different domain and genre of texts. Evaluating the content organization in the medical domain is impracticable, and therefore, we did not evaluate this module intrinsically. We, instead, rely on the extrinsic evaluation reported in Chapter 6. In the news domain, we conducted a human evaluation of the ordering strategy.

4.4.1 Ordering in TAS

Given the difficulty to recruit medically trained subjects to read and evaluate summaries, we searched for an automatic evaluation for this component. Recent advances in automatic assessment of cohesion (Barzilay and Lapata, 2005) provide us with a readily available evaluation framework. One of the advantages of this scoring method is that it is syntactically, not lexically, based. Thus, even though it was trained on news data, it potentially could be applied to texts with different vocabularies, such as the texts summarized by TAS. However, Barzilay and Lapata (2005) report that their model's accuracy degrades considerably (by 20%) when tested on a domain other than the one it was trained on and, thus, advise to retrain the model for each new domain. Because we do not have any examples of summaries annotated with readability measures, however, we could not use this evaluation method.

There are two other methods published for automatically evaluating the ordering of a machine-generated text. Lapata (2003) proposes Kendall's coefficient as a metric. It counts the number of permutations needed to go from a given ordering to an ideal ordering. Since we do not have access to ideal summaries and their

orderings, we cannot apply this method. Furthermore, as we have shown, there is no single ideal ordering, and comparing one ordering against only another single valid ordering might not be the most reliable way to asses the quality of an ordering strategy. More recently, "sentence continuity" (Okazaki, Matsuo, and Ishizuka, 2004) was proposed as part of the evaluation of ordering for multi-document summarization. A score is computed by counting n-gram overlap from one sentence to the following one. Since such a measure rewards repetition of information, we decided to ignore it: TAS aims to identify repetitions and merge them in order to present concise information.

4.4.2 Ordering in News Summarization

We compared our ordering strategy to two baselines strategies: a simple "chronological ordering," where every piece of information is ordered individually based on its publication date, and a "majority ordering," where a linear ordering among pieces of information is found such that the agreement among the orderings provided by the input articles is maximized.[5]

We collected a corpus of 25 sets of articles. For each set, we produced three summaries different only in their ordering strategy: one summary was produced using the Majority Ordering, one using the Chronological Ordering and one using the Block Ordering. Thus, there were 75 summaries overall. We asked three human judges to grade the quality of the order of information in each summary. The manual effort needed to compare and judge system output is extensive considering that each human judge had to read three summaries for each input set as well as

[5]For more information on the majority ordering and its implementation, see Barzilay, Elhadad, and McKeown (2002).

	Poor	Fair	Good
Chronological Ordering	10	8	7
Majority Ordering	3	14	8
Block Ordering	3	8	14

Table 4.1: Evaluation of different ordering strategies in news summarization.

skim the input texts to verify that no misleading information was introduced in the summaries. Judges were asked to grade a summary as Poor, Fair, or Good. We used an operational definition of a Poor summary as a text whose readability would be significantly improved by reordering its sentences. A Fair summary is a text which makes sense, but reordering of some sentences can yield a better readability. Finally, a summary which cannot be further improved by any sentence reordering is considered a Good summary.

The judges were asked to grade the summaries taking into account only the order in which the information is presented. To help them focus on this aspect of the texts and not how they had been generated, we bypassed the generation component of MultiGen and presented a sentence extraction version of the summary. In addition, we manually resolved any dangling references beforehand.

Table 4.1 shows the grades assigned to the summaries. We are showing here the majority grade that is selected by at least two judges. This was made possible because in our experiments, judges had strong agreement; they never gave three different grades to a summary. Block Ordering provides a significant improvement in the quality of the orderings, both from the Chronological Ordering and the Majority Ordering ($p=0.04$ and $p=0.07$ respectively using a Fisher exact test and conflating Poor and Fair summaries into one category to obtain a 2x2 table).

4.5 Related Work

The task of content organization is relevant to both text summarization and text generation.

In single-document text summarization, summary sentences typically are arranged in the same order that they were found in the full document, although Jing and McKeown (2000) report that human summarizers do sometimes change the original order. When it comes to multi-document summarization, the summary consists of fragments of text or sentences that were selected from different texts. Thus, there is no complete ordering of summary sentences that can be found in the original documents. Yet most summarizers have no explicit organization component. In the news domain, information is ordered according to the publication date of the input articles it was extracted from (Radev, 1999; Lin and Hovy, 2002; Schiffman, 2005). Our work on news summarization, as well as more recent work adopting the idea of ordering by block (Okazaki, Matsuo, and Ishizuka, 2004), shows that an explicit ordering strategy, which relies on topical relatedness to group clusters of information first, produces more readable summaries.

In domain-dependent summarization and text generation, it is possible to establish valid orderings *a priori*. For instance, in the specific domain of news on the topic of terrorist attacks, texts can be constructed by first describing the place of the attack, followed by the number of casualties, who the possible perpetrators are, etc. A valid ordering of information is traditionally derived from a manual analysis of a corpus of texts in the domain, and it typically operates over a set of semantic concepts. In text generation, a top-down approach, using schemas (McKeown, 1985) or plans (Dale, 1992) relies on such a methodology to determine the organizational

structure of the text. This approach postulates a rhetorical structure which can be used to select information from an underlying knowledge base. Because the domain is limited, an encoding can be developed of the kinds of propositional content that matches rhetorical elements of the schema or plan, thereby allowing content to be selected and ordered. Rhetorical Structure Theory (RST) allows for more flexibility in ordering content by establishing relations between pairs of propositions. Constraints based on intention (Moore and Paris, 1993), plan-like conventions (Hovy, 1993), or stylistic constraints (Bouayad-Agha, Power, and Scott, 2000) are used as preconditions on the plan operators containing RST relations to determine when a relation is used and how it is ordered with respect to other relations. In contrast, our approach to text organization is bottom-up: content items are dynamically grouped together based on their similarity.

While traditionally content ordering rules were identified through manual analysis of a corpus of target texts, there has been more research done recently to induce such rules from examples of target texts. While some approaches require semantic annotation of the target texts (Duboue and McKeown, 2001; Kan and McKeown, 2002), others learn ordering rules in an unsupervised fashion (Lapata, 2003; Barzilay and Lee, 2004). However, so far, unsupervised methods work best when the corpus to analyze is formulaic, and there are few irregularities in the organization of the target texts. One of the difficulties in our case, as for most multi-document summarization systems, is to find target summaries from which to learn. To our knowledge, there is no available corpus of human-written summaries that synthesize results from different clinical studies.

Finally, another distinction between our approach and most of the methods

adopted by most generation systems is architectural. Our ordering component takes place after the content selection of the information in a pipeline architecture, in contrast to generation systems, where ordering and content selection usually come in tandem in the phase of content planning. This separation might come at a cost: if there is no good ordering to the given extracted information, it is not possible to backtrack to the content selection module and extract new information. However in summarization, content selection is driven by salience criteria. We believe that the same ordering strategy should work with different content selectors, independently of their salience criteria. Therefore, we prefer to keep the two components, selection and ordering, as two separate modules.

Chapter 5

Content Generation

This chapter describes how the summarizer transforms a set of Content Items, as selected by the Content Organization stage, into an English sentence.

Traditionally, in full-fledged generation systems, sentence planning and lexical choice modules handle generation of complex structures. In our framework, however, we do not aim, at this stage, to generate multiple syntactic and lexical variations but, rather, to convey summary content in a straightforward fashion. As such, we chose a phrasal generation approach to the generation process. We take advantage of the fact that our input is textual – in contrast to the semantic input of generation systems – to re-use input phrases instead of relying on manually built grammars and lexicons.

Because it re-uses phrases from the input articles, the resulting text is necessarily targeted at readers comfortable with the language of clinical studies. The class-based user modeling is carried out only once the text summary is generated as described in the next chapter.

We first describe our strategy for mapping Content Items to English sen-

tences by relying on a library of generation templates. We then present the lexical choice for the Concepts contained in the Content Items and explain how references are added to each sentence in the summary so that the user can at any point access the original input information.

5.1 Sentence Generation

As we have defined the types of messages we are interested in summarizing, namely results, we have designed a semantic representation for them in the form of Content Items. We took advantage of the formulaic nature of the language of clinical studies to report results and, thus, employed information extraction patterns to extract them. In an equivalent fashion, we want now to be able to convey the summary content in English sentences. Since there are few types of messages, it makes sense to employ a phrasal generation approach to the sentence generation task. Such an approach has advantages: generation templates are used, and so there is no need for an actual syntactic realization module – that is, the constituents of the sentence are already ordered, and their syntactic category is fixed in the template. In addition, the realization of the semantic slots is now independent of the way the sentence as a whole is realized. Paraphrasing power is not derived at the semantic level but from encoding as many templates as needed. While phrasal generation might not provide the most varied output and does not scale up to larger generation problems, it was shown to be a successful approach for several generation systems (Kukich, 1983; Hovy, 1988; McKeown, Kukich, and Shaw, 1994).

We determined a sublanguage to convey the target messages in the form of generation templates (Kittredge and Lehrberger, 1983). Semantic slots are then

plugged in within one of the generation templates. At the end of our organization stage, a subset of Content Items to be generated into a sentence might contain a singleton (one Content Item to generate), repetitive Content Items, aggregated Content Items, or contradictory Content Items. We describe the generation of these messages in turn.

5.1.1 Generation of Non-Contradictory Content Items

In contrast to generation systems, our input is textual in nature. We take advantage of this fact and re-use the extraction patterns used at the content selection stage, this time as generation templates. The sublanguage for the generation of non-contradictory Content Items consists of the extraction patterns that instantiated the most Content Items when run against a development set of clinical studies. Since there are six Relations – association, risk, prediction, absence of association, absence of risk, and absence of prediction – we have six libraries of generation templates. Based on the Relation of a given Content Item, the generation module selects a generation template to use from among the corresponding templates. Figure 5.1 shows examples of generation templates. The first four patterns and the last one realize a prediction Relation, and the four middle ones an association Relation. The text, as well as the syntactic role for the semantic slot Parameters and Findings (NP, or noun phrase, and NG, or noun group, that is, a conjunction of noun phrases), is directly taken from its corresponding extraction pattern.

While sentence generation is conceptually independent from the generation choices of the previous sentences in the summary, we incorporate two inter-sentential factors in our generation process. First, to introduce variety, we keep

track of which templates were recently used in the text and avoid re-using them twice in a row. Second, we attempt, where possible, to maintain a consistent focus from one sentence to another (McKeown, 1985). For the first summary sentence, the focus is approximated as the terms used in the input user query: if the Parameters of the Content Items to be realized are mentioned in the user query, the Parameters should be the focus of the sentence and, thus, should be realized before the Findings. Conversely, if the Findings mention query terms, then they are realized first. For each subsequent sentence, we keep track of the focus of the previous sentence and, if possible, maintain it through ordering Parameters and Findings. The order in which the Parameters and Findings must appear to satisfy this criterion necessarily constrains our choice of templates to the ones that present this slots in the correct order.[1] For instance, the first four templates in Figure 5.1 put the focus on the Parameters, while the last one focuses on Findings. If after applying these two constraints, more than one template is available to us, we choose one randomly.

At the conclusion of the content organization stage, we know which Content Items should be realized within the same sentence, based on whether they can be conceptually aggregated. The only aggregation phenomena handled by TAS are conjunction and ellipsis.[2] Our design is very similar in spirit to the strategy used by the PlanDoc generation system (McKeown, Kukich, and Shaw, 1994). We perform aggregation in the following fashion: when the Content Items share the same Concept (whether Parameters or Findings), ellipsis is used to achieve a concise

[1]This constraint may limit our ability to vary which templates are selected from one sentence to the next, depending on the number of templates used for generation.

[2]The topic of aggregation is complex and covers many phenomena. See Reape and Mellish (1999) for a detailed review.

Analysis identified [PARAMETERS/NG] as predictors of [FINDINGS/NP]
Analysis identified [PARAMETERS/NP] as a predictor of [FINDINGS/NP]
Analysis identified [PARAMETERS/NG] as predictors of [FINDINGS/NG]
Analysis identified [PARAMETERS/NP] as a predictor of [FINDINGS/NG]
[PARAMETERS/NG] were related to [FINDINGS/NP]
[PARAMETERS/NP] was related to [FINDINGS/NP]
[PARAMETERS/NG] were related to [FINDINGS/NG]
[PARAMETERS/NP] was related to [FINDINGS/NG]
Predictors of [FINDINGS] included [PARAMETERS]

Figure 5.1: Three Generation templates examples, each with their possible variations depending on the number of Parameters and Findings.

sentence. The other Concepts are merged as a conjunction of noun phrases, each noun phrase verbalizing a given Concept.

5.1.2 Generation of Contradictory Content Items

In our summaries, which report results from multiple studies, contradiction may arise. In our corpus of clinical studies, however, there are no examples of sentences reporting contradictory results. As such, we have no readily available language to re-use to verbalize contradictions. In attempting to address this problem, we manually analyzed a small corpus of review articles in cardiology, hoping to find such messages, but could not find any such sentences per se.[3] We, therefore, created a simple single generation template to signal to the reader the presence of a contradiction. Given any two contradictory Content Items, the summary sentence will be verbalized as the first sentence in Figure 5.2, where [CONTENT_ITEM1] and [CONTENT_ITEM2] are verbalized as described above for regular Content Items.

[3]Review articles present information from different clinical trials.

[REF1] reports that [CONTENT_ITEM1], but [REF2] reports that [CONTENT_ITEM2].
[REF1,REF2,REF3] report that [CONTENT_ITEM1], but [REF4,REF5] report that [CONTENT_ITEM2].

Figure 5.2: Generation templates for contradictions.

When one or both of the Content Items involved in the contradiction are merged repetitions, we use a similar template, but with the appropriate number of references. For instance, if the first Content Item is merged from three repetitions and the second is merged from two, the contradiction will look like the second sentence in Figure 5.2. The actual references ([REF]) are verbalized as described in Section 5.3.

5.2 Lexical Choice for Concepts

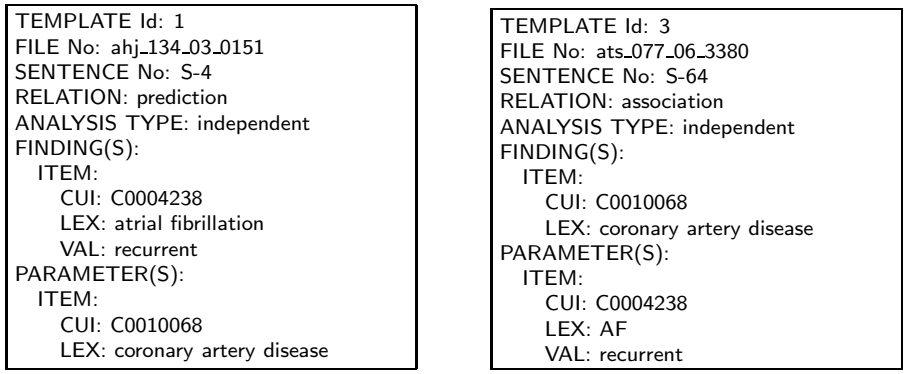
The task of lexical choice is, given a semantic entry, to choose the best verbalization among different alternatives. Traditionally, lexical choice is a complex task, which interacts with the sentence planning module. In our framework, however, because we employ a phrasal generation approach, the only decision left at this stage is how to verbalize the slots in the generation template. The template controls the syntactic role of each slot (Parameters and Findings). In our case, both Parameters and Findings are always verbalized as noun phrases. When there is only one Concept, whether Parameter or Finding, it is verbalized as a noun phrase. When the slot is an aggregation of several Concepts, they are verbalized as a conjunction of noun phrases.

Lexical choice relies on a dictionary, which provides semantic entries and their alternate verbalizations to choose from. Building such a dictionary is a bottleneck to most generation applications, as it is typically manually compiled. Since TAS is a summarizer, its input is not semantic in nature, as in generation systems. We, therefore, can take advantage of the input verbalizations to build our lexicon on the fly, as we select content.

When there has been no aggregation or merging at the sentence level, that is, when only one Content Item contributes to a given sentence, the generation of the Parameters and Findings is simple: we use the original phrase extracted at the Content Selection stage and stored in the `ORI` field of each Concept. This way, the verbalization of each Concept is as close as possible in meaning to the original text.

When multiple Content Items contribute to the sentence, either because of repetition or aggregation, several Concepts with the same `CUI` must be merged and verbalized as one noun phrase. Consider the two Content Items in Figure 5.3, which will be verbalized as a merged repetition. An issue arises because the common Finding is verbalized differently in each of the two Content Items. In the first Content Item it is verbalized as "recurrent atrial fibrillation," while in the second it appears as "recurrent AF." The values in both Concepts are the same ("recurrent"), but the lexemes are different. In such instances, because we aim to produce a summary, we simply choose the shortest lexeme. In this case, the Finding will be verbalized as "recurrent AF."

There is an additional wrinkle introduced when several Concepts with the same `CUI`, but different associated values, underlie a single generated noun phrase. For instance, one Concept, e.g., atrial fibrillation, might have one associated value



(a) (b)

Figure 5.3: Two Content Items to be merged.

"pre-operative" and another "recurrent." Another set of Concepts, for instance "left atrial size," may have two different associated values "> 4.0cm" and "< 60mm." As discussed in Section 3.1.1, matching values to determine their similarity is a research challenge. In light of this fact, it would be misleading to choose one value among them to verbalize in the generated text, as it would imply that all the Concepts in question share the same value. Instead, we choose not to convey any value whatsoever, and verbalize only the lexeme. While this strategy may introduce occasional distortions, such distortions are, to an extent, inevitable given the very project of summarization. The reader is expected to read further about any interesting reported result by accessing the input studies, to which references appear in the text summary.

5.3 Generation of References

The summary in TAS acts as an access point for the user to identify the relevant results in the input studies and decide from there which studies to read in more

> Predictors of in-hospital mortality included smaller aortic anulus, age , prior stroke , coronary artery disease , serum creatinine > 1.3 mg / dL , NYHA , congestive heart failure , hypertension, preoperative atrial fibrillation , ejection fraction < 40 % and concomitant CABG [1,1,2,2,2,2].

Figure 5.4: Summary sentence and references.

detail. Once a set of Content Items is realized into a sentence, the Content Items are gathered once again, and their metadata, i.e., the `ARTICLE` and `SENTENCE` slots, are collected to provide the references for the summary sentence. For instance, the summary sentence in Figure 5.4 contains six references, to two sentences in the first article and to four different sentences in the second. Each reference is a link to the actual sentence from which the Content Item was extracted.

Once the summary is displayed, TAS looks up which of the input articles actually contributed to the summary content. For each such article, title and publishing information is displayed. The remaining articles are also listed, under the heading "Not referenced." This way, the user can still access them by clicking on their titles. Appendix A provides an example of a generated summary.

Chapter 6

Extrinsic Evaluation

This chapter reports on the overall evaluation of the text summarizer. We first review the different evaluation methods present in the literature. We then present the results of a task-based evaluation conducted with physicians.

6.1 Evaluation Methods for Text Summarization

As the field of text summarization has evolved, the question of how to evaluate summaries has become more and more important. Evaluation of summarization is still very much an open research question. As with other machine-generated texts, several aspects of a summary can be assessed, from its content to its organization to the language it uses. Content selection strategies can be evaluated by comparing summaries to one or several ideal summaries, whether automatically (Lin and Hovy, 2003) or manually (DUC, 2002-2005; van Halteren and Teufel, 2003; Nenkova and Passonneau, 2004). Automatic methods for measuring the quality of a sentence (Bangalore, Rambow, and Whittaker, 2000) and, more recently, for

assessing the overall coherence of a text (Barzilay and Lapata, 2005) have been proposed. One defining hypothesis underlying summarization is that summaries provide the reader with an added value over the original texts, whether it manifests itself as time gained in reading a summary instead of the original set of input texts or an understanding of the information superior to the one the user would have gotten by reading the input documents alone. Task-based evaluations help verify this hypothesis, and confirm other assumptions, such as the context in which the summarizer will be used (Spärck-Jones, 1999). In a task-based evaluation, the summarizer is evaluated in a black-box fashion. Given a task, and a suitable metric, task performance is measured using the summarizer in lieu of the full-text input documents.

Several tasks have been looked at, from human relevance judgments (Jing et al., 1998; Tombros and Sanderson, 1998; Mani et al., 1999; Kushniruk et al., 2002), to more complex ones, such as text classification (Mani et al., 1999), text comprehension (Morris, Kasper, and Adams, 1992; Maybury, 1995), and report writing (McKeown et al., 2005). In the relevance judgment task, subjects are typically shown either summaries or actual documents, whereupon they decide whether the text is relevant to a given topic. For the classification task, subjects choose the most representative class under which a summary or a document should be categorized. The subjects' performance is defined as the number of correct decisions they make and is compared against a gold standard (e.g., the TREC collection in the large-scale TIPSTER SUMMAC evaluation (Mani et al., 1999)). For the reading comprehension task, subjects answer a multiple-choice questionnaire after reading either summaries, the documents themselves, or without the benefit of any prior

supporting text whatsoever. The task performance is measured as the number of correct answers given. For the evaluation of the Newsblaster system (McKeown et al., 2005), users were asked to write a report to answer a question about an issue in the news. They had access to either only the source documents, the documents and machine-generated summaries, or the documents with human-written summaries. In this case, assessing the subjects' performance is not as straightforward as in the previous tasks. Since there is no ideal report to compare the subjects' reports to, the comparison is made against a composite report built from the reports of all subjects combined. In most task-based evaluations, other measures are collected in addition to the subjects' performance. Time-to-task-completion, for instance, is a good indicator of the added value of presenting summaries instead of source documents to the user.

6.2 Task-based Evaluation with Physicians

In our evaluation we want to determine whether our summarizer is helpful to physicians at the point of patient care. More precisely, we test two hypotheses: (1) summaries help users access relevant information; and (2) personalized, generated summaries are better than generic, extractive summaries in accessing relevant information. In a real-world situation, a physician treating a patient rarely has the time to read the salient medical literature in detail, yet knowing about findings relevant to the patient under care can help the physician decide on an appropriate treatment. For physicians in training, it is also important to learn to pinpoint quickly the findings relevant to a given scenario or to a real patient and to decide on a course of action accordingly. As concluded in the course of a preliminary

feasibility study, it is not enough for a physician to know that a study pertains "globally" to a patient; some findings are particularly relevant, while others are less interesting (McKeown, Jordan, and Hatzivassiloglou, 1998). Consequently, we define "accessing relevant information" as being able to identify the findings that *pertain to a particular patient in a given set of clinical studies*. Accordingly, we designed a task where subjects must select such findings in a limited amount of time and under three different conditions: (1) given a set of studies and with no other material, (2) with a generic summary, or (3) with a personalized summary. Performance and subjective assessment of the task allow us to determine to what extent summaries help access relevant information given the same set of input articles. Our focus is not on evaluating whether or not the input articles themselves are relevant for a scenario, as our summarizer has no control over its input. We first describe the methods (evaluation design and analysis criteria). We then report on results and discuss the main findings of this evaluation.[1]

6.3 Methods

6.3.1 Evaluation Design

For their task, subjects were asked to select findings relevant to a given scenario. Each scenario in the study was constructed in the following way. We provided our medical expert with an anonymized patient record and asked him to suggest an interesting open-ended question for this patient. In particular, we asked the expert to think of questions similar to the ones posed to a physician-in-training when

[1]This section is based on an AMIA paper (Elhadad et al., 2005).

presented with a new patient to treat. We would then search for clinical studies on the medical search engine PubMed by submitting several queries relevant to the question and patient's characteristics. We then presented the expert with ten to 15 articles returned by our search. The expert selected among them four to five articles relevant to the patient and question to be included in the scenario. In addition, in order for the input articles to simulate the typical results of a search engine, we asked the expert to select two to three articles less relevant to the patient and question from the articles with which we provided him.[2] So, overall a scenario, as presented to each subject, consists of the latest discharge report of a patient record, a clinical question about the patient and a set of seven to eight pre-selected clinical studies. We used three different scenarios: (1) a female patient with atrial fibrillation, (2) a male patient with atherosclerosis of the saphenous vein, and (3) a male patient who must undergo aortic valve replacement.

Each subject was asked to read the patient record and then select from the input articles any findings that are relevant to the patient and question. We chose the granularity of a "finding" to be a sentence, so that subjects were instructed to select a whole sentence whenever it contained a relevant finding. In practice, one sentence does not always correspond to a single finding, and one sentence can sometimes report several findings at once. However, we believe that a sentence is a good approximation, in part because it is a more natural concept to explain to subjects with no linguistic background (as opposed to selecting clauses or phrases within a sentence), and in part because of the difficulties entailed in using an automatic tool to identify such clauses or phrases.

[2]We need the expert to control for the input articles, as we focus on evaluating the summarization, not the search for relevant articles.

To make the task more realistic, we asked subjects to complete the task for each scenario in at most 15 minutes. Under this time constraint, subjects, unlike our expert, did not have time to read each input article in its entirety and were forced to figure out a searching and reading strategy to help them identify relevant information efficiently. This task simulates what physicians, and especially physicians-in-training, would typically do when presented with a new case: look for answers to their questions in the literature, but in a very short amount of time, given the pressure of working at the point of patient care.

We designed our own interface that allowed subjects simply to click on a sentence in any input article to select it. The selected sentences are automatically displayed in a separate window, which is always visible and groups all the selections from the different articles, and are also highlighted inside each article. Articles are color-coded, so that the subject knows at any time which sentence came from which article. A screen shot is shown in Figure 6.1. The interface was tested in a pilot study with one physician for the same task as the one used in this study. It is written as a set of Java scripts and automatically generated html pages and runs within the Firefox web browser.

Even though the interface was designed with our study in mind, we kept the interface task-independent. It is a general annotation tool for selecting information from a set of articles into a single space (the Selections window). Our goal was to allow users to select information and manage their selections in an efficient way. As such, we designed it so that as much information as possible is displayed on the screen, and the users do not spend too much time navigating among different windows and can, therefore, focus on their selection task. In addition, presenting

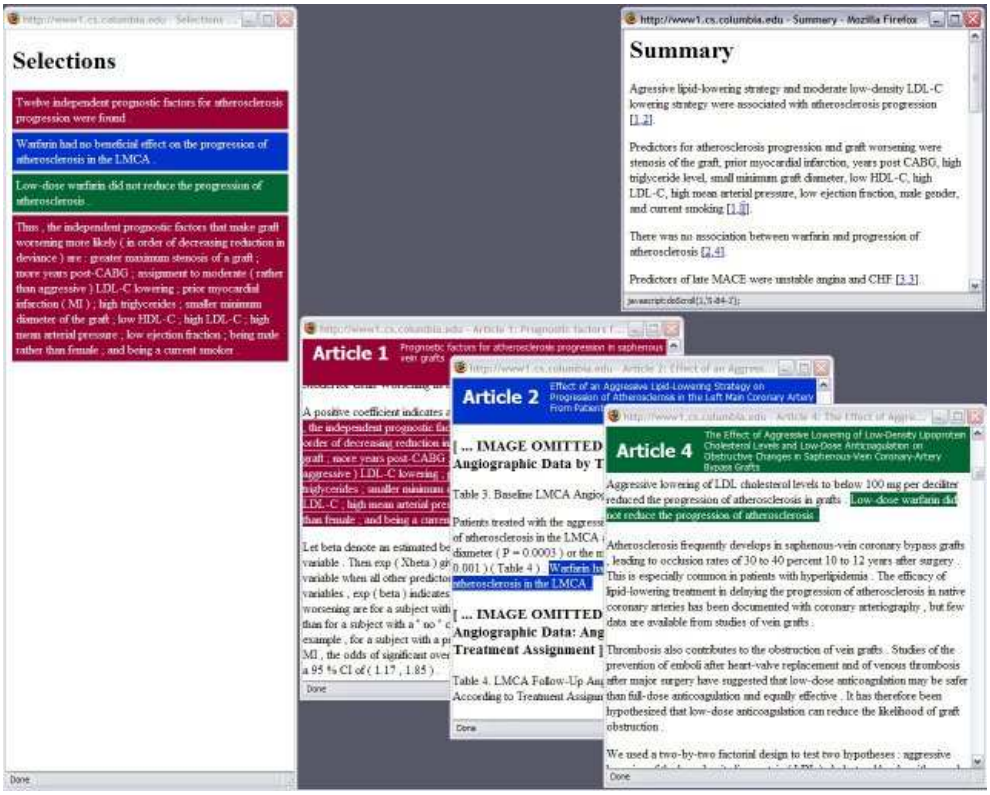


Figure 6.1: Screenshot of the user interface designed for the study.

the same interface to the subjects, no matter the conditions under which they are being tested, allows us to control the environment in which the subjects interact and to focus on the effect of each testing condition.

At the beginning of the study, the subjects were shown a demo of the interface, and were given instructions on the task, with examples of findings and non-findings. The interface, the instructions, and the scenarios are available at `http://www.cs.columbia.edu/~noemie/tas_eval`.

We compared our summarizer to two baselines. In the news domain, one competitive baseline is to present the lead sentence of a document (DUC, 2002-

2005). This baseline, while highly effective in news summarization, is not competitive for summarizing scientific articles, since the first sentence usually conveys background information rather than the gist of the article. Another possible baseline for our task would be to present the abstracts, written by the articles' authors. But, since each scenario contains eight input articles, we decided that eight abstracts would be too lengthy for the reader to look at. (Moreover, abstracts are readily available within each input article in any event.) Instead, we chose a more competitive baseline in the form of a generic, extractive summary that did not provide personalized content. The summaries were made generic by turning off the content personalization module in TAS. Because extractive summaries are standard among summarization systems and, as such, constitute a fair baseline, we decided to make the summaries extractive and not generated, as they are in the full version of TAS. We also compared the subjects' performance when presented with no summary at all, as when using a regular search engine. Thus, each subject was tested under the three following conditions.

Search A list of articles was provided to the subject, in the same format as the results of a search engine. For each article, title, publishing journal and year of publication were displayed.

Generic A generic, extractive summary (see example in Appendix D) was provided to the subject. The references in the summary were pointers to specific sentences in the input articles that were extracted and included in the summary. The summary was followed by a list of articles in the same format as in the Search condition. If the summarizer had not selected any sentences from a given article, that article was still listed for the user under a heading called

"Non-referred Articles."

Personalized A personalized, generated summary (see example in Appendix D) was provided to the subject. The references in the summary were pointers to specific sentences in the input articles from which information in the summary was drawn. References had the same function as in the Generic condition. The summary was followed by a list of the articles that were referenced by the summary and the optional list of non-referred articles.

Each medical scenario was presented to each subject under one of the three conditions. The order in which conditions were presented was systematically varied. Thus, the first subject saw Scenario 1 first under the Search condition, followed by Scenario 2 under the Generic condition, and Scenario 3 under the Personalized condition, while the second subject saw Scenario 1 first under the Generic condition, etc.

After selecting the findings for a given scenario, the subjects were asked to fill out a questionnaire (see Figure 6.2). The questions were answered on a five-point scale, with a high grade indicating a positive answer.[3] While the first four questions are about the scenario difficulty and the interface in general, Q5 through Q8 are about how the summary facilitated the task. Q9 was a free-text question for the subject to report any comment on the summary. Therefore, when tested under the Search condition, the subjects were asked to answer only questions Q1 to Q4.

[3]For Q5, a high grade indicated that the sentences in the articles were accessed mostly through the references inside the summary.

Question
Q1. Did you feel like you had enough time to identify all the relevant findings?
Q2. At the end of the task, do you think you have a reasonable answer?
Q3. Did you feel that the interface was helpful in supporting your task?
Q4. If you had the opportunity, would you use this interface again?
Q5. How did you access the articles?
Q6. Did you read the summary?
Q7. Did you feel that the summary saved you time?
Q8. Did you feel that the summary content was relevant to the given question, patient under care and studies?

Figure 6.2: Questionnaire given to the subjects after each completed scenario.

6.3.2 Analysis

We conducted two types of analysis: an objective measurement of how well the subjects did on the task in terms of quality of results, as well as a subjective evaluation, which measures subject satisfaction with the system.

Objective Evaluation

To quantify each subject's performance on the task, before the study began, we collected a gold standard set of findings for each scenario. Our medical expert read each of the articles on paper and highlighted the sentences which conveyed findings pertinent to the patient, following the same guidelines as the ones given to the subjects in the study. We imposed no time limit on the collection of the gold standard. Overall, the expert took approximately four hours to select the relevant findings in the 23 articles contained in the three scenarios (so roughly an average of ten minutes per article).

During the study, the interface stored which sentences were selected by the subject, as well as timing information. We scored a subject's selection in terms of

its precision and recall as compared to the gold standard. We combined precision and recall into the F_2 measure, which equally weights precision and recall using the harmonic sum of the two numbers. Previous work on evaluation of summaries has expressed concerns about strictly comparing a set of selected sentences against a gold-standard selection (Jing et al., 1998). One troublecase scenario may arise where the gold standard contains one sentence, and a subject selects a different sentence, yet one that conveys the same fact. A direct sentence comparison would count this as a mismatch. We took this consideration into account in our study, and instructed the expert responsible for the gold standard and our subjects to select every relevant finding, even when it was a repetition of an already selected finding.

We used the Kruskal-Wallis one way analysis of variance to determine the impact of the testing conditions on the performance of the subjects, as measured by F_2. This test is well suited to our problem, as it allows us to determine whether the three independent samples are from different populations. It is a non-parametric alternative to the ANOVA test and, therefore, makes fewer assumptions about the nature of the data (Siegel and Castellan, 1988). We also looked at the effect of scenario and subject to verify their possible impact.

Subjective Evaluation

We relied on both the post-scenario questionnaire filled out by the subjects and the video transcript of the subjects to assess user satisfaction with interacting with the interface under the different conditions. We used the Kruskal-Wallis one way analysis of variance and the Wilcoxon-Mann-Whitney test to analyze the question-

naire answers. For the video, we asked subjects to think out loud and performed cognitive analysis on their behaviors (Kaufman et al., 2003). Such analysis allows us to decipher how the subjects react to the study. In particular, we were interested in finding out their degrees of satisfaction under each condition. This is also captured by the post-study questionnaire, but examining what the subjects say during the course of the task allows us to gather more details about the reasons for their satisfaction or dissatisfaction. In addition, such analysis provides insight into how subjects adapt to the task under each condition and whether they develop reading strategies to enhance their performance. Data used in the analysis included an annotated and time-stamped transcript from the audio tape, video captures of the screen, and the selected findings. The transcript was coded for actions, goals, inferences about the articles or scenarios, comments about the system, and expressions of uncertainty.

6.4 Results

Twelve subjects were recruited from the Cardiac Intensive Care Unit at New York Presbyterian Hospital, ranging from a fourth-year medical student to an attending physician for 15 years. The subjects were financially compensated. We report on the results for only 11 subjects out of the twelve recruited. Six subjects completed the full study, one subject completed two scenarios out of three, and four subjects completed only one scenario out of the three. One subject – the one whose results we do not report – was interrupted in the middle of his first scenario and could not come back at all for the other scenarios. These failures to complete the study were mainly due to the difficult conditions of the ICU that day: some subjects were

interrupted in the middle of a scenario. We dropped the results of the scenarios that were not finished. Thus, overall we have 24 data points from 11 subjects. From this point on we only refer to these scenarios.[4]

6.4.1 Task Performance Analysis

All subjects but one spent all 15 minutes under every condition. We understand this result as a confirmation that the alloted time was a constraint, making the task more difficult to perform accurately without the benefit of a reading strategy. On average, the generic summaries contained 36.6 sentences, and the personalized summaries contained 8.3 sentences. The articles themselves contained each on average 127.7 sentences (and each scenario comprised seven to eight articles).

Table 6.1 shows the mean F_2 measure by condition across the three scenarios. When presented with the personalized, generated summaries, the subjects performed the best. Their performance decreased if they were presented with a search interface with no summary. They performed the worst when presented with generic, extractive summaries. The Kruskal-Wallis analysis and post-analysis comparisons applied to the different conditions showed that the condition influenced the subjects' performance (p=0.08) (see Figure 6.2). The difference between the Search and Personalized conditions, and the Generic and Personalized conditions were both statistically significant (p=0.05). Interestingly, although the overall number of selected findings was similar for the Generic and Personalized conditions (mean of 12.9 vs. 14.2, not statistically significant), the selections made under the Personalized

[4]The results presented in this thesis are different from the ones reported in the AMIA paper (Elhadad et al., 2005): we report here the results from 11 subjects, while the AMIA paper reported the results for only the six subjects who had completed the whole study.

Condition	F_2 Mean
Generic	12.4
Search	14.6
Personalized	22.2

Table 6.1: Mean subject performance per condition.

condition were far more accurate. Similar results occur with the Search condition (mean number of selected findings was 12.625). This confirms our hypothesis that personalized summaries are objectively helpful for users in the context of a task.

We also evaluated the influence of the two other factors entered in the model: scenario and subject (see Figure 6.2). For the scenarios, Kruskal-Wallis analysis indicated that at least one of the scenarios had an effect on the subjects' performance (p=0.07). Post-analysis revealed that subjects performed significantly worse when tested under Scenario 1 than when tested under the two other scenarios (mean F_2 of 11.4 compared to 22.2 for Scenario 2 and 16.4 for Scenario 3). As we explained in Section 3.5.2, Scenario 1 was a difficult scenario, which required general knowledge of cardiology. Since most subjects were physicians-in-training without a specialty in cardiology, it makes sense that they did not perform well on this particular scenario.

Finally, we evaluated the influence of individual subjects on our data. For this analysis, we evaluated only the six subjects who completed all scenarios. Kruskal-Wallis analysis shows that there were no significant differences among individual subjects with respect to their performance (p=0.34).

	Df	Kruskal-Wallis chi-squared	p
condition	2	5.04	0.08
scenario	2	5.30	0.07
subject	5	5.57	0.34

Table 6.2: Kruskal-Wallis one way analysis of variance for condition, subject, and scenario.

Question	Personalized	Generic	Search
Q1. Did you feel like you had enough time to identify all the relevant findings?	3.25	3	3
Q2. At the end of the task, do you think you have a reasonable answer?	3.6	3.1	3.2
Q3. Did you feel that the interface was helpful in supporting your task?	4.7	3.9	4.4
Q4. If you had the opportunity, would you use this interface again?	4.1	4	3.9
Q5. How did you access the articles?	2.9	1.7	NA
Q6. Did you read the summary?	3.9	2.1	NA
Q7. Did you feel that the summary saved you time?	3.7	2.6	NA
Q8. Did you feel that the summary content was relevant to the given question, patient under care and studies?	4.2	2.8	NA

Figure 6.3: Questionnaire given to the subjects after each completed scenario and the average responses per condition.

6.4.2 Questionnaire Analysis

The average answers to the questionnaire are given for each condition in Figure 6.3. While the first five questions are about the scenario difficulty and the interface in general, Q5 through Q8 are concerned with whether the summary facilitated the task. The subjects had a positive reaction to the interface, independently of the condition. Accordingly, the answers for Q1 to Q4 are not significantly different across conditions. The subjects found the interface very helpful for the task (Q3

had an overall average grade of 4.3 out of 5), and they reported that they would enthusiastically use the interface again (the average overall grade for Q4 was 4 out of 5). We take this result as a validation of our interface.

The subjects showed a strong difference of opinion between the Personalized and Generic conditions with regard to the summary questions (Q5 to Q8). Overall, subjects were not satisfied with Generic summaries. Q5 tested how much of the information in a summary was exploited by the subjects. A higher grade means that the subject relied heavily upon the summary as an access point to get to a specific sentence in an input article directly. Subjects relied on Personalized summaries more than they did on Generic summaries (p=0.05). The subjects reported reading very little material from the Generic summaries, while they read more than half of the Personalized summaries (see the grades for Q6, p=0.03). The subjects found that, overall, the Personalized summaries saved them time in their task, while they felt that the Generic summaries made them waste some time (marginally significant, p=0.07). Finally, the subjects found the Personalized summaries to be highly relevant to the scenarios, but not the Generic summaries (p=0.01). All this confirms that, subjectively, the subjects were satisfied with Personalized summaries but were not content with the Generic summaries.

6.4.3 Cognitively Based Video Analysis

We illustrate the cognitive analysis by comparing a subject on two scenarios. In Scenario 1, the subject is presented with a Generic summary and in Scenario 3, she is given a Personalized summary. Figure 6.4 shows excerpts from Scenario 1,

as categorized and characterized through cognitive analysis techniques.[5]

00:25 Action: Open and Reviews Summary
Comment: Some of it doesn't seem as relevant to the actual treatment options, What's the best treatment? Yeah. So, some of this stuff seems more descriptive.
[...]
06:53 Action: Clicks on Article 3 from reference list.
07:05 Action: Selects sentence to be added to list.
07:38 Action: Clicks on article 5 from reference list
Comment: I'm kind of getting bogged down in the summary, in terms of this woman's presenting, she's now in afib, so we're looking for more of a cardioversion treatment. So going right to the list, some of these articles are talking more about maintenance.

Figure 6.4: Excerpts from a subject looking at a generic, extracted summary.

Early on in the process, the subject finds the Generic summary to be largely uninformative and unrelated to her goal of determining the best treatment options. After a few minutes, the subject abandons the text summary and goes straight to the article index. Her approach is not systematic. This is in marked contrast to her performance when using the personalized summary as indicated in the excerpt shown in Figure 6.5.

When using the personalized summary, this particular subject develops an effective reading strategy characterized by the following action pattern: (a) review pertinent paragraph in summary, (b) click on indexed article (from summary), (c) select sentences to add to selection list, (d) repeat (c) until all relevant sentences from the article have been added, (e) shift focus back to summary, (f) select new article or same article with new entry point. The Personalized summary allows her to identify candidate articles easily and to go directly to the relevant passages

[5]The example discussed in this section was annotated and analyzed by Dave Kaufman (Kaufman et al., 2003).

00:54 Action: Opens Summary, immediately goes to list of index articles.
2:33 Clicks/Selects Article 1 from summary
Comment: From here I'll give the summary a try and see if I can figure out how to use it.
2:45 Selects sentence from article 1
3:11 Selects sentence from article 1
4:37 Selects article 2 from summary
4:56 Selects sentence from article 2
Comment: Moving on to article two. Nicely brings you, actually, to the main results sentence which is a great summary.
11:32 Selects Article 5 from summary
11:46 Selects sentence from article 5
11:52 Selects sentence from article 5
Comment: All seem to be relevant and in support of the findings.

Figure 6.5: Excerpts from the same subject looking at a personalized, generated summary.

or sections within the articles through the links of the summary. The result is a greater number of correctly selected sentences. In addition, she expresses greater satisfaction with results. Some representative comments from the subjects while they performed the task are presented and discussed in the section below.

6.5 Discussion

Although the interface was designed for this evaluation only, the subjects enjoyed its main functionalities, and in particular the presence of the Selection window to copy the important findings. As one subject commented, "it's nice to see this and be able to take out different important parts of it and separate it. It's almost like taking notes. You look over to the left hand side of the screen and you can kind of

read it in bullet point, which is nice."

Being presented with a summary was a mixed blessing for the subjects. As one subject summed it up: "If the summary is well written it's better to have the summary. Otherwise it doesn't help at all."

Personalized summaries allowed the subjects to complete the task more successfully and with greater satisfaction than under other conditions. Subjects noticed the Personalized summaries and reacted to them positively: "It does save some of my time because it helped me to focus right away on what the question was." said one subject. Another concluded: "I think the summary was good. I think the articles were way too long and some of them were not appropriate. [...] The articles were definitely more time consuming and didn't give me as much info as the summary." The feature of the summaries that links to specific sentences inside the articles was overall very well received by the subjects. The identification and merging of repetitions was also noticed and appreciated by most subjects: "Some of this is sort of redundant but it's good: the same information, just a different point in the article."

Contrary to our expectations, we found that generic summarization did not improve access to information. Subjects did not like the generic summaries and, in most cases, gave up reading them to focus on the list of articles provided after the summary. The subjects complained about two aspects of the summaries: sentence extraction and lack of personalization. Since the summaries were produced using sentence extraction, some summaries contained dangling references, which may have been confusing to some subjects. However, the comments of many subjects indicated to us that an abundance of irrelevant information in the summary,

i.e., lack of personalization, contributed greatly to their dissatisfaction with generic summaries. Following are two representative comments reflecting this issue:

> "I didn't feel it was relevant to the question because it didn't identify for me why the patients had aortic valve replacement and it's talking about aortic stenosis and there's nothing in my discharge summary that's telling me that's why my patient had the aortic valve replacement." Here the subject complains that there is a mismatch between the information in the patient record and the findings reported in the summary.

> "The first line is not very relevant to my question. I don't even know what they're talking about. [...] Okay, we're going to skip this. I didn't find the summary helpful at all with this patient so I'm just going to look at the article."

In addition to this issue, subjects were put off by the length of generic summaries. Both the lack of personalization and the use of sentence extraction may contribute to making summaries long: there is no filtering of irrelevant information, and the sentences are not fused but concatenated.

The subjects did not particularly complain about seeing the articles under the Search condition, which presented typical search engine results. This could well be due to the fact that users are used to such interfaces, and while they are not optimal, they have adapted to such engines and developed their own reading strategies over time to find relevant information amid many documents. "Interesting. It's a totally different format, instead of just doing Medline." commented one subject,

showing that, despite the differences in the interface, he has recognized that the Search condition is nothing other than a typical search engine, like the one he is accustomed to using. This is now a familiar situation.

This feeling of familiarity may also explain why subjects performed better overall under the Search condition than under the Generic condition. In the time allotted to perform the task under the Generic condition, the users spent quite some time trying to make sense of the summary as a tool to access information. When, given the multiple issues inherent to the Generic summaries, they did not succeed in doing so, they tended to abandon the summaries and use only the list of articles, thereby reverting to a more familiar search strategy.

In some cases, however, subjects having had the benefit of a summary in connection with a prior scenario, recognized the inefficiencies of their accustomed strategy when tested under the Search condition. One subject complained for instance:

> "I would have felt more comfortable on the summary thing if it had brought you right to the key points in the results section, I wouldn't really need to go back and read the study. I would have had enough to go on and not feel totally lost. Now I'm finding myself just kind of like searching through... still talking about data stuff, how they recorded data, kind of a waste of time. This is not very efficient."

There were some issues in the design of the study. First, the subjects seemed unwilling to accept the fact that the input articles were pre-selected for them. We chose to pre-select the articles in order to isolate the effect of summarization on the task. We did not want to evaluate subjects' search strategies. Although we

stressed this point in the instructions, this remained a source of confusion, and some subjects stressed the importance of choosing your own input articles. "I'm not sure I would use it because the value of this really depends on the articles it links you with."

There was low agreement among the subjects on selection of the findings, even for the subjects tested under the same condition for the same scenarios. Our analysis shows that this is due in part to repetitions across articles. When information was repeated, some subjects selected each instance and others only one of the repeated findings. We had asked subjects in the guidelines to select all instances (as we had asked our expert when building the gold standard), but this did not come across for most subjects (the guidelines are shown in Appendix D).

In an ideal setting, it would be good to evaluate the effect of generation and the effect of personalization separately. However, doing so would require additional subjects to be recruited. Getting physicians to sit down for enough time to conduct a study is notably difficult to achieve. Despite a high monetary compensation for participating in our study, we were able to recruit only a limited number of subjects. Physicians just do not have the time.

Finally, we asked the subjects to perform the task and think aloud, both during the 15 minutes allotted to the task. The act of thinking aloud can be an additional cognitive load on the subjects and may have conceivably affected their performance on the task, though none of the subjects so indicated. Ideally we would have conducted a separate study with the goal to specifically analyze the subjects' interaction with the system through a think-aloud protocol. Practically, however, it was difficult to recruit enough subjects for two separate studies.

Chapter 7

Technical to Lay Adaptation

In this chapter we describe our strategy for adapting a text originally targeted at medically knowledgeable readers into a text understandable by a lay reader.

From an architectural standpoint, catering the text to another level of expertise as an editing of an already generated text, as opposed to a semantic input, is an attractive strategy. A method designed to modify a machine-generated text can later be applied to other texts, not necessarily machine-generated.

We first present an analysis of the language used in technical and lay texts at the document level, the sentence level and the vocabulary level and show that, while technical texts are distinguished from lay texts in several ways, word usage is both the most amenable to automated intervention and likely to have a drastic impact on user comprehension.

We, therefore, focus on improving the comprehensibility of technical terms and show in a feasibility study that determining whether a given term is likely to be understood by a lay reader can be achieved by examining the term in isolation, rather than necessarily in the context of a given text. Our method is a two-stage

approach: (1) identify in a given technical sentence which terms are likely to be incomprehensible to a lay reader, and (2) supplement the complex terms with mined definitions. Evaluation shows that our automatic method yields a significant improvement in reader comprehension.

7.1 Data Collection and Processing

In this section we describe the corpus we rely on for the task of adaptation: medical texts written for consumers of health information. We refer to it in the remainder of this chapter as the ReutersHealth corpus.

The corpus contains news stories summarizing clinical studies from the Reuters Health E-line newsfeed. Reuters journalists take technical publications and report the main findings, methods and sometimes interviews with the authors of the publication. There are two important characteristics of this corpus: (1) the stories are written for a lay audience, and (2) every story in our corpus contains a reference to the original publication. This way, we can gather examples of texts conveying the same information but written for different audiences. The stories draw upon studies from reputable medical journals, such as Annals of Internal Medicine, New England Journal of Medicine and Lancet. Among them are ten cardiology journals that are typically reported on by Reuters journalists.[1] Appendix E shows an example of ReutersHealth story.

There is no way to collect all the stories from Reuters Health E-line newsfeed, so we mined the stories written about specific clinical trials. We did so by searching

[1]Journal of the American College of Cardiology, American Heart Journal, American Journal of Cardiology, American Journal of Hypertension, Arteriosclerosis Thrombosis and Vascular Biology, Heart, Circulation, Hypertension, International Journal of Cardiology, and Stroke.

Number of texts	9,775
Number of sentences	160,208
Number of words	4,373,104

Table 7.1: ReutersHealth corpus statistics.

.

for stories containing a SOURCE reference. Overall, we mined such stories from 2002 to October 2005. After removing duplicate stories, our corpus contains 9,775 texts. Table 7.1 shows statistics about the corpus.

Our adaptation technique also relies on information from the MRC Psycholinguistic Database (Coltheart, 1981). This database contains 150,937 words of general English with up to 26 linguistic and psycholinguistic attributes for each. Among them, a subset of words is indexed with a familiarity index. It also contains words from the Brown corpus, which are annotated with their frequency counts (Kucera and Francis, 1967).

7.2 Differences in Technical and Lay Languages: an Analysis

In this section we provide a qualitative analysis of the differences we identified between the texts in our technical corpus of clinical studies and the ReutersHealth stories. We identified differences at the document level, at the sentence level, and at the level of vocabulary usage.

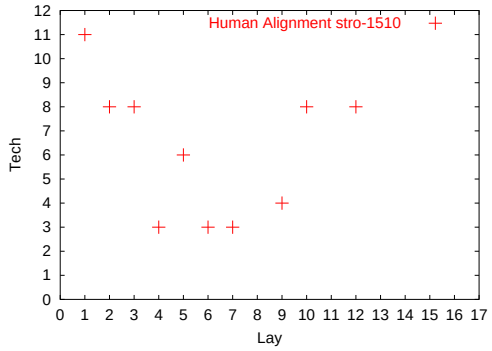


Figure 7.1: Manual alignments for a ReutersHealth story and a corresponding technical abstract. A dot on (x,y) indicates that sentences x and y were aligned.

7.2.1 Document-Level Differences

We manually aligned the sentences in a subset of ReutersHealth stories with the abstract of the corresponding clinical study they report on. A sentence in the lay story and one in the technical abstract were aligned if they conveyed the same information. We aligned 55 pairs of texts. Figure 7.1 plots the manually aligned sentences for one of the text pairs. Notably, the order in which the information is presented in the technical abstract is not followed by the ReutersHealth story. While abstracts typically present background information, methods, results and conclusions, the ReutersHealth stories tend to present conclusions first, followed by background. Methods are often described, although not each time, and finally results of the study are reported. Explanations of medical concepts are mentioned at every point, as well as interviews with the authors of the study. This change of discourse structure from the technical texts to the lay texts confirms what previous research had observed (McKeown, 1985; Paris, 1993; Barzilay, Elhadad, and McKeown, 2002).

The presence of these organizational regularities indicates that the readers

probably expect such ordering of information, whether because these features inherently enhance comprehension or simply because readers have become accustomed to this format. An application which simplifies a technical text should take this fact into account. In our framework, we focus on sentence adaptation, and thus, we do not address this problem.

7.2.2 Sentence-Level Differences

It is difficult to quantify the differences in syntactic structures from technical sentences to lay sentences because parsers do not perform accurately on technical medical data. There is no doubt, however, that some sentences in technical texts have specific structures, which are not seen in general English sentences. Consider the sentence "The age-adjusted RRs corresponding to intakes of < 0.5 drinks/day, 0.5 to 2 drinks/day and > 2 drinks/day were 0.76 (95% confidence interval: [CI]: 0.52 to 1.12), 0.64 (95% CI: 0.40 to 1.02) and 0.59 (95% CI: 0.32 to 1.09), respectively, as compared with nondrinkers (p for TREND=0.06)." This content of this sentence was rendered by a ReutersHealth journalist as "Compared with nondrinkers, men who consumed half a drink or less per day cut their heart disease risk by 24%, while those who drank one-half to two drinks daily cut their risk by 36%." The structure of the technical sentence is unique in the sense that it verbalizes as a whole the concept of risk ratio and cannot be broken down into smaller conceptual semantic units. While conveying the same information, the lay version of this sentence employs a far more conventional structure.

These differences in structure are distinct from the ones discussed in the literature on syntactic simplification. There, the challenge is to identify and modify

the individual clauses contained in a sentence. Tackling sentence level simplification in our framework, however, would require drastic reworkings of the sentence as a whole. As in the example above, clauses cannot be restructured in isolation.

To qualify the differences in sentence structure between technical medical texts and their lay counterparts, we withheld a test set of 1,885 sentences and trained a trigram language model on the remaining texts from the ReutersHealth corpus. The language model was trained with classes of words, where terms tagged by our term identifier were substituted with the generic token "TERM."[2] This allows us to get beyond differences in vocabulary from one type of text to the other. We compare the perplexity of the test set against a set of technical sentences drawn from abstracts of clinical studies. The technical sentences were grouped under background sentences, methods, results and conclusion.[3]

Table 7.2 shows the average perplexity and out-of-vocabulary counts of different sets of sentences run against the language model. As expected, the lay sentences from the withheld ReutersHealth set are the closest to the lay texts used for training (with a low perplexity of 59.6). We can see that some types of sentences in the technical texts have simpler structures than others. For instance, the background sentences are the closest to the lay sentences. In contrast, the methods, results and conclusion sentences are the further away from the lay language. Because we use a class-based language model, which minimizes the vocabulary issue, we attribute the changes in perplexity to different sentence structure.

[2]We used the Kneser Ney discounting as implemented by the SRI language modeling toolkit.

[3]The grouping was carried out by selecting studies with structured abstracts and categorizing the sentences according to their headers.

	Average perplexity	OOV
1,885 sentences ReutersHealth sentences	59.6	228
395 background technical sentences	95	143
306 conclusions technical sentences	104	102
691 results technical sentences	133.3	865
452 methods technical sentences	145.5	291

Table 7.2: Average perplexities against a class-based lay language model.

	Average perplexity	OOV
1,885 sentences ReutersHealth sentences	96	297
395 background technical sentences	245	285
306 conclusions technical sentences	251.7	194
691 results technical sentences	264.1	1060
452 methods technical sentences	302.4	444

Table 7.3: Average perplexities against a lexicalized lay language model.

7.2.3 Vocabulary-Level Differences

We verify here that medical jargon is more frequent in technical texts than in texts targeted at lay readers by training a fully lexicalized language model on the same set as described above. The average perplexities and the number of out-of-vocabulary (OOV) words are shown in Figure 7.3. Both are higher than in the class-based version of the language model, but more importantly, the increase in perplexity is much wider from the lay sentences to, for instance, the methods sentences. We attribute this to the high number of out-of-vocabulary words and conclude from this that there are major vocabulary differences between technical and lay texts.

Our work focuses on the differences of vocabulary from technical to lay texts. Identifying which words in a sentence are hard to comprehend for a reader is not

a trivial task. While typical medical texts contain many technical terms, not all of them need to be explained; many are sufficiently familiar to lay readers. We turn to the issue of distinguishing the terms in need of explanation from those that do not require explication in the next section.

7.3 Predicting Comprehensibility of Vocabulary

We address the task of predicting the lay reader's ability to understand a given term without having access to or requiring any explicit cognitive model of the reader. Instead, we approach class-based personalization as a text-to-text project. We rely on our knowledge of the properties of texts targeted at lay readers – and hence, putatively comprehensible to such readers – to predict automatically the terms that a lay reader would be likely to understand and, by extension, those too difficult for the lay reader. The texts collections we use are both in the medical domain and out-of-domain, containing general English language.

One important question in this regard is whether or not context plays a significant role in determining the reader's ability to understand a given term. A study we conducted with three subjects indicates that context is, in fact, not a significant factor. As the scope of our study was limited to the technical medical domain, which contains jargon and terms very unfamiliar to most readers, we are not in position to conclude one way or the other whether context would be similarly insignificant in the comprehension of terms in sentences from other domains, including general English sentences.

We first describe this study, followed by our method to identify complex terms and our evaluation results.

7.3.1 Comprehensibility and Context

To investigate the role of context in understanding terms in medical texts, we asked three lay readers to classify terms as comprehensible or incomprehensible by looking at a list of 100 out-of-context distinct medical terms. We repeated this experiment with the same subjects a week later, this time providing the same terms in context.

Our subjects had a college education but no special medical knowledge. The 100 terms were identified by our term identifier and randomly selected from result sentences from our corpus of clinical studies. They ranged from those that would be familiar to nearly all lay readers, e.g., "illness" or "sleep deprivation" to those that only medical professionals would be likely to recognize, e.g., "transient ischemic attacks" or "PTCA."

For the out-of-context list, the subjects were asked to circle the terms which they did not understand. For the in-context list, they were presented with 100 technical sentences containing the 100 terms highlighted and were asked to circle the highlighted terms that they did not understand.

For the out-of-context list, the subjects exhibited almost-perfect agreement in their annotation (Kappa of 0.83). In all, 53 of the 100 terms were judged comprehensible and 47 incomprehensible. For the in-context list, subjects were in substantial agreement (Kappa of 0.76). The slight dip from the agreement of the out-of-context scenario may be attributable to which context unsettles individuals' certainty about whether they understood a given term. Nonetheless, both levels of agreements show that there is wide consensus among subjects as to which terms are too technical and which ones are familiar and understandable.

More importantly, each of the subjects exhibited substantial agreement with

their own earlier out-of-context judgments (0.79 for two subjects, 0.76 for the third). That is, each subject made substantially the same decisions about whether or not they understood a term's meaning *regardless* of whether the term appeared in or out of context.

There were a few notable exceptions where context obviously made a difference, however. For example, the term "HRT," not understood out of context, became quite clear in the sentence "Information on hormone replacement therapy (HRT) in women of the 6th and 7th decades was obtained retrospectively." Conversely, it is interesting to note that, where disagreement appeared between out-of-context and in-context judgments, the transitions were not always from not knowing a word to coming to understand its likely meaning in context. In some cases, the introduction of a context actually made the subjects aware that a word they thought they knew was being used in an unfamiliar fashion. For instance, "compensation," which all subjects found comprehensible out of context was then judged incomprehensible by two subjects in the sentence "Partial left ventriculectomy can provide structural remodelng of the heart that may result in temporary improvement in clinical compensation."

We conclude from this study that examining a term in isolation is an appropriate strategy to adopt for determining its comprehensibility. We describe next our algorithm.

7.3.2 Methods

Psycholinguistic research has shown that frequency of word usage in a large corpus is a good predictor of its familiarity. High frequency words are usually found to

elicit a higher recognition than low frequency words (Forster, 1976; McClelland and Rumelhart, 1981; Morton, 1969). Our operative assumption to decide whether a term is likely understandable follows these findings.

Knowing that the ReutersHealth articles are targeted at a lay audience, we conclude that frequent terms in the ReutersHealth corpus are likely to be understood by a lay reader. We define the frequency of a given term as the sum of the frequencies of its morphological variants (e.g., "stroke" and "strokes" would be counted as two occurrences of the same term). Whenever a term is above a pre-determined threshold, it is considered comprehensible. To gather accurate frequency counts for all terms, including multi-words ones, we ran the texts in ReutersHealth through our term identifier and handled any term as an individual token.

In addition to the in-domain knowledge gathered from the ReutersHealth corpus, we investigated the use of general English resources to help us prune out familiar words. We used the Brown corpus (Kucera and Francis, 1967) for this purpose. In contrast to ReutersHealth, the words contained in the Brown corpus are very unlikely to be medical in nature. When tested on our development set, we found that considering all words with frequency count higher than one as comprehensible provides the best results.[4]

Besides relying on frequency count, we investigated whether the familiarity index of a word, when available, can predict its comprehensibility accurately. We tried to incorporate the familiarity index provided by the MRC Psycholinguistic

[4]The development set was obtained by asking a human lay judge to annotate a randomized set of 100 terms, other than those used in our initial study. Of these terms, 60 were judged comprehensible and 40 incomprehensible.

Database. However, their list was too small for our purposes. Polysemy has been found to be another predictor of word familiarity (Jastrzembski, 1981). WordNet, for instance, uses polysemy count as an index of familiarity for a word. We tested its use to prune out familiar words on our development set, but it did not yield satisfactory results. Even words as universally comprehensible as "adult" were found unfamiliar by WordNet because of its low polysemy count. The addition of either resource to our technique did not improve results on the development set.

In addition, we institute a rule that automatically classifies all abbreviations as incomprehensible to a lay reader. We do so due to our observation that each of the abbreviations occurring in our development set was classified as incomprehensible by our subject. We choose to implement a high-precision rule to treat them all as incomprehensible because even lay articles will occasionally make liberal use of such abbreviations after first defining them for the lay reader. Consequently, we could not rely on our frequency measure alone to classify a substantial majority of the abbreviations correctly.

7.3.3 Evaluation

To test our method, we used the out-of-context list of terms annotated by our three subjects for our initial study as a gold-standard. We looked to a majority of our subjects to conclude whether a given term on our list was comprehensible or not to a lay reader.

As a comparison baseline, we implemented a classifier based on hard-coded rules and specific to the medical domain: a term is classified as incomprehensible if its UMLS semantic type is among the following: diseases, therapies, drugs, chem-

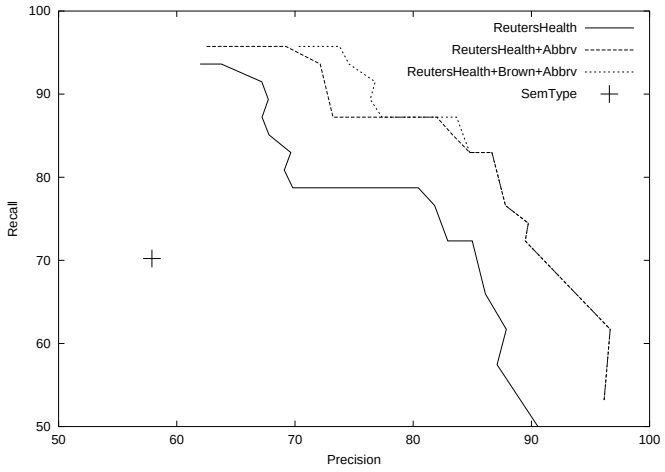


Figure 7.2: Precision/recall for the SemType method, our full method and variants.

icals or pathological functions. For example, the term "valvular regurgitation," classified as a pathological function, would be judged incomprehensible, while "alcohol use," classified as a behavior, would be judged understandable to the lay reader. We refer to this baseline as SemType.

Figure 7.2 shows the precision/recall curve for (1) the SemType baseline, (2) a variant of our method that relies on ReutersHealth only, (3) a variant relying solely on ReutersHealth and our abbreviation rule and (4) our full method, which makes use of ReutersHealth, our abbreviation rule and the Brown corpus. Precision and recall counts are based on the identification of incomprehensible terms as compared to the gold standard. The curves are obtained by varying the frequency threshold on the ReutersHealth corpus.

Our SemType baseline, a single point on the graph, yields a decent 57.9% precision and 70.2% recall. In contrast, our full method yields around 90% precision for the same recall level. This confirms that most medical terms incomprehensible

to the lay reader are indeed likely to come from among the semantic types we selected (diseases, therapies, drugs, chemicals or pathological functions), but that SemType is ultimately not sufficiently sensitive to distinctions between familiar and unfamiliar terms within these semantic types and fails entirely to account for unfamiliar terms outside these types. In addition, SemType relies on the UMLS, and thus, is domain-dependent.

Relying on ReutersHealth frequency alone provides a better alternative. The addition of the abbreviation rule improves precision at all levels of recall. At high levels of recall, the Brown corpus is a useful supplement for identifying and pruning out very familiar words.

In our framework, we are primarily interested in high recall strategy. A slightly lower precision triggers a few more terms than necessary to be categorized as incomprehensible and in need of elucidation. This would not likely impede a reader's ability to understand the resulting text. In addition, it could have the benefit of sweeping in terms on the threshold of the average lay reader's comprehension and outside the comprehension of some lay readers. On the other hand, a low recall would leave many technical terms undefined, thereby sacrificing comprehension.

When applied to our development set, we determine the frequency threshold in ReutersHealth of 7 to provide the best compromise between high recall and good precision. On our test set, this yields a precision of 86.7% and 83% recall.

7.4 Augmenting Technical Sentences

Once the terms that are too complex for lay readers are identified in a given technical sentence by the previous step, the next task is to adapt the terms to make

them more comprehensible.

One attractive solution would be to substitute a simpler synonym for a complex term. While appealing, this solution is difficult if not impossible in the medical domain. Although a small subset of medical terms have clear synonyms that make their meanings apparent to a lay audience (e.g., "varicella" can be substituted with "chickenpox," no matter the surrounding context),[5] the vast majority of the medical terms in our corpus do not fall into this category. There is no easy way to provide exact synonyms for terms such as "enzyme inhibitor," "heparin," or "left ventricular ejection fraction."

Rather, such terms require a paraphrase, a definition or an explication. This is the approach adopted by the journalists of the ReutersHealth stories. For instance, the term "left ventricular dysfunction" is never mentioned in the ReutersHealth corpus; however, the concept is discussed and referred to as: "patients whose heart muscle is weakened, as measured by left ventricular ejection fraction (LVEF), a test of the heart's pumping ability." An ideal solution would be to substitute (in context) automatically acquired lay paraphrases for complex technical terms. However, paraphrase acquisition is still very much a research challenge and has never been attempted in the medical domain. In Section 8.4, we describe a first step to acquire such paraphrases, namely, sentence alignment. As we will explain later, because there is much lexical variability from technical to lay texts, our results are insufficiently accurate to apply them in practice to the task of paraphrase acquisition.

Our adaptation strategy, therefore, relies on definitions. We supplement sen-

[5]Although replacing "varicella" in the above sentence would not be a good idea."

tences with definitions for each complex term. Other terms, the ones not detected as complex, are left as is. When simplifying a set of sentences, we make sure to define only the first mention of a complex term. Inserting definitions into the text itself, either as clauses immediately following the unfamiliar term or as sentences following the one in which the term appears, is impractical because multiple complex terms can and often do occur within a single sentence and because definitions can be long and unwieldy. The fluidity of the text would suffer too greatly from such an approach. For this reason, we chose to provide definitions as links associated with complex terms. Proceeding in this way both preserves the integrity of the text and permits readers who do not feel the need to refer to a definition for a particular term to read on without distraction.

To gather definitions, we collected several medical glossaries, either general or specific to cardiology. We found, however, that none had sufficient coverage to be useful for our task when tested on a development set of sentences. Consequently, we relied on Google's "define:" feature to obtain definitions. Using Google is advantageous because the work of mining definitions from multiple glossaries and web pages is already done for us.

Because Google returns multiple definitions, which can be from any domain, and because terms, especially abbreviations, can have different meanings in different domains, we select the shortest definition (in number of words) returned that defines the given term in a medical context. To determine whether a given definition is medical in nature, we run it though our term identifier and count the ratio of terms recognized by the UMLS (and, therefore, more likely to pertain to the medical domain). High-ratio definitions are considered medical in nature. This

way, "BP" gets properly defined as "abbreviation for blood pressure" rather than "before present" or "British Petroleum."

Where a multi-word term does not return a definition, we remove the leftmost word and search again until we identify a phrase or word, if any, for which a definition can be found. We make clear to the user what portion of the term is being defined. Where we can find no definition whatsoever, we provide none.

Besides augmenting terms with definitions, we implemented a simple cosmetic editing of technical sentences to remove all parenthetical phrases. We did this because such parentheticals in medical texts almost always contain statistical information or specification that would likely be both incomprehensible and distracting to a lay reader: e.g., "In analyses adjusted for age, sex, and smoking, intake of wine on a monthly, weekly, or daily basis was associated with a lower risk of stroke compared with no wine intake (monthly: relative risk [RR], 0.83; 95% CI, 0.69 to 0.98; weekly: RR, 0.59; 95% CI, 0.45 to 0.77; daily: RR, 0.70; 95% CI, 0.46 to 1.00)."

7.4.1 Evaluation

We describe our evaluation setup and results of adapting a technical sentence for lay readers.

Our test set consisted of 89 sentences taken from our technical corpus of clinical studies. We selected only sentences which convey results, as these are the ones our summarizer processes. For each sentence, we identified automatically the terms in need of definition using our method described in the previous section. Overall, the 89 sentences contained 256 terms, as defined by our term identifier.

Among the 256 terms, 98 were identified as too complex for lay readers. Out of the 89 sentences, 24 sentences did not contain any term considered complex. This means that for these, no further adaptation could be carried out. We kept the remaining 65 sentences in our test set.

For each of the 98 terms, we gathered two sets of definitions: one automatically from Google, as described above, and a gold standard obtained under the supervision of a physician. We presented our physician with the technical sentences and terms to be defined highlighted and gave her a list of possible definitions for each term. The list was the original Google list. We asked the physician to select the one that she thought defined the term best for a lay reader. If she did not find any satisfactory definition, she was free to supply her own.

To evaluate both the extent to which definitions improve comprehensibility of technical sentences for lay readers and the extent to which our strategy is successful in providing useful definitions, we presented a subject with three sentences at a time: a technical sentence, followed by our automatically augmented sentence, followed, in turn, by the sentence with gold-standard definitions. The subject was asked to read the technical sentence and rate it on a 1-5 comprehensibility scale (5 being most comprehensible). The subject was then asked to rate, in turn, the automatically augmented sentence and, finally, the gold-standard sentence.

Table 7.4 shows the mean comprehensibility ratings for the three versions of the 65 test sentences. Technical sentences alone were rated 2.23 on average. Sentences augmented with gold-standard definitions yielded the best rating (4.26), which is close to full comprehension. Our automatic method lies between the two, with a 3.72 average rating, significantly improving reader comprehension of the

	Comprehensibility rating (1-5)
No definition	2.23
Automatic definitions	3.72
Gold-standard definitions	4.26

Table 7.4: Mean comprehensibility rating for the technical sentences, the sentences provided with automatic definitions, and for sentences with ideal definitions.

technical sentences ($p<0.0001$ under a Wilcoxon-Man-Whitney test).

The quality of our automatically supplied definitions affects the overall quality of our method, and this evaluation indirectly evaluates their quality. Examples of good definitions included the one for "ACE inhibitor:" "A drug that makes the heart's work easier by blocking chemicals that constricts capillaries" and the one for "bolus:" "a single dose of drug." Abbreviations were often associated with wrong definitions, because they are highly ambiguous. This is true even within the medical domain. For example, "AIS" was defined as "Androgen Insensitivity Syndrome" whereas our physician defined it as "arterial ischemic stroke," because she understood the context surrounding the term. Similarly, EF, which in our cardiology subdomain stands for "ejection fraction" was defined as "Epilepsy Foundation." Finally some definitions in our test set suffered from the drawbacks of Google's mining techniques: "cholesterol" was defined as "One large egg contains 213 mg cholesterol."

We understand the increase in comprehension from the technical sentences to the augmented sentences as a validation that vocabulary is an essential gateway to comprehension and that defining unfamiliar terms can make incomprehensible sentences comprehensible. 30.7% of the test sentences went from a comprehension

level of 1 or 2 to a 4 or 5 level when automatic definitions were provided. The increase was even more pronounced when sentences were augmented with gold-standard definitions (47.7%). Moreover, in all but one of the ten cases when neither the automated nor the gold-standard definitions improved the comprehensibility of sentences, this result was most likely due to the fact that the terms designated incomprehensible by our automatic method were, in fact, comprehensible terms (e.g., “image quality” and “anticoagulation treatment”).

Chapter 8

Conclusions and Future Work

We conclude this thesis by a summary of the main contributions of our work. We also discuss the main limitations of our work and propose directions for future work. In particular, we focus on the task of text simplification and describe our work on sentence alignment for comparable monolingual corpora. Such a technique could prove useful for learning rules to translate a technical sentence into a lay version.

8.1 Contributions

We presented a novel approach to text summarization that takes the user into account by producing summaries sensitive to the user's points of interests as well as his/her level of expertise. Following is a summary of the contributions of this work, both at the functional level and at the technical level.

8.1.1 Functional contributions

- **Design and implementation of a multi-document summarizer of clinical studies.** The research conducted for this thesis was incorporated in the multi-document summarizer TAS. TAS is a framework to test personalization strategies. It takes as input several clinical studies returned from a search engine and generates a single summary of the findings relevant to a user model. TAS is part of the PERSIVAL medical library (McKeown et al., 2001).

- **Tailoring to users' individual characteristics.** One of the goals of this work is to demonstrate that relying on information about the users, such as their interests, helps them access relevant information more accurately. Using an existing patient record as a proxy for a user model, we designed an algorithm that selects the findings pertaining to a patient's characteristics. We investigated whether personalized summaries can benefit from an existing, non-structured user model. Our extrinsic evaluation shows that personalization is indeed a helpful feature when it comes to accessing relevant information.

- **Tailoring to users' class-based characteristics.** In addition to incorporate the interests of a user, we investigated how to take into account the user's background. Our work focuses on the terminology of technical texts. Relying on texts targeted at lay readers, we determine which terms are too technical to be understood by a typical health consumer. Such terms are supplemented with an appropriate definition. Our study with subjects shows that provid-

ing definitions of complex medical terms improves sentence comprehension significantly, and confirms that medical terminology is a major obstacle to understanding.

- **Tools for medical text processing.** The medical domain offers a very attractive opportunity to investigate important issues in user-sensitive summarization. On one hand, physicians deal with many different patients but need to find information relevant to each. On the other hand, patients have little medical knowledge but still want to be informed of the most current medical information pertaining to their conditions. The work presented in this thesis contributes to the field of medical NLP. In addition to providing a summarizer that addresses the needs of these two populations, we implemented a series of tools to process medical articles: a classifier to determine the genre of a publication in a medical journal, a classifier to determine the clinical task of a clinical study, an efficient document parser which identifies the structure of a technical article (sections, subsections, headers, etc), tags each words with their parts of speech, and identifies medical terms along with their associated values.

8.1.2 Technical contributions

- **Content representation.** Using information extraction techniques and relying on the UMLS ontology, the summarizer operates over a data structure that is between full semantic analysis and extracted text. Content Items allow for easy implementation of the filtering strategies and content organization strategies.

- **Dynamic content organization.** A coherent organization of the summary content is approximated by clustering the Content Items based on their semantic similarity. The ordering relies on an automatically assigned priority weight for each cluster. Ordering, in contrast to most work in generation, is dynamic and bottom-up, which allows for more flexible texts. In addition, the rationale for our algorithm – ordering by blocks – is domain and genre independent. So far, our algorithm has been applied to both summarization of medical clinical studies and of domain-independent news stories.

- **Text summary re-generation.** The summary sentences are produced by reusing the extracted phrases from the input articles and incorporating them in a phrasal generation framework. The generation templates themselves are reused from the patterns written for the content selection stage. Language reuse is an attractive hybrid between sentence extraction and deep generation. Unlike in the sentence extraction paradigm, it is possible aggregate information from different sources into a single fluent sentence. Yet, the tedious task of manually constructing the entries in a lexical chooser, like in traditional generation systems, is not needed anymore; the concepts are realized in whatever ways they were realized in the input texts.

- **Text-to-text adaptation.** Class-based personalization is modeled as a text-to-text problem. We focus on adapting a technical sentence at the lexical level, and more specifically at the term level. We conducted a study that indicates that a reader's comprehension of a term is similar whether the term is presented in context or not. Our strategy to determine which terms are unlikely to be understood by a lay reader relies on statistics from corpora

targeted at consumers of health information. Supplementing complex terms with definitions mined from existing glossaries provides a significant increase in reader's comprehension.

- **Collection of resources for text-to-text generation.** For our work on text-to-text adaptation, we collected corpora of texts which convey similar information but are written for different audiences. Such texts allow us to study similarities and differences from one style to another. One corpus, Britannica, contains regular encyclopedia articles for cities paired with the same entries in a different version targeted at young adults. Another corpus is in the medical domain and contains abstracts of clinical studies paired with Reuters news articles summarizing the studies for consumers of health information. A significant portion of each corpus was manually aligned at the sentence level. We hope that these resources will help research in text-to-text generation and paraphrasing. Annotated sentences from the Britannica corpus have already been used by other researchers, notably in the task of textual entailment (Dagan, Glickman, and Magnini, 2005).

8.2 Limitations

Portability. TAS is a domain-dependent summarizer. For TAS to be ported to a different domain, the following is required:

- **User model** – TAS expects an existing user model which contains information that is directly comparable to the information conveyed in the input texts.

- **Ontology** – TAS relies on the UMLS as the ontology for the domain. The CUIs provide an abstraction over and above the Content Items and the user model and allow for concept matching (both between two Content Items and between a Content Item and the user model).

- **Degree of abstraction** – the type of information to be conveyed in the summary can be represented by a data structure, such that at least shallow semantics can be automatically obtained to allow for further processing (content filtering and content organization).

Patient characteristics representation. While there are many advantages to relying on the patient record – it is obtained in a non-intrusive way and it contains lots of accurate pieces of information about a patient – there are also challenges. One of them is that not all the pieces of information are equally important for the personalization strategy. In our implementation, we represent the patient record as a flat list of terms and their associated values. We do not attempt to build a more complex representation of the user. Even though our evaluation of the summarizer shows that personalization is possible with a naive representation of the patient, we believe that the summarizer would strongly benefit from a more sophisticated representation of the patient characteristics.

User model representation. Our user model encodes two types of users: physicians and patients. This, however, is a coarse distinction. Within the patient population, different patients have widely different levels of expertise. This can be due to their prior knowledge or to how invested the patients are in their own medical history and how much they educate themselves about it. Within the physician

population, physicians are likely to approach a case with different sets of questions depending on their specialty and years of experience. An anesthesiologist will not care about the etiology of a disease, while this might be a very important question for an internist. A resident may be interested in basic information that is already known by a more experienced doctor. A realistic user model should take these distinctions into account. In our framework, users can ask a query to direct the search for input articles, thereby specifying their personal interests in the case. The summarizer takes the query terms into account, but ideally a more complex representation of the users' interests could be achieved. Our decision to keep a "shallow" user model is in sync with our choice not to make any inferences about the cognitive state of the users or the information to be summarized.

Ontological knowledge. TAS relies on the UMLS ontology to identify medical terms and their associated semantic types (disease, body part, etc.). However, besides that degree of abstraction, TAS does not take further advantage of the knowledge encoded in the UMLS. We chose not to integrate any ontological information from the UMLS because it is too noisy and not sufficiently consistent.[1] Nonetheless, our similarity metric could benefit from additional ontological knowledge: if two Concepts are siblings, for example, they should be more similar than two unrelated Concepts.

Content Representation and salience criterion. We defined our basic representation, the Content Item, as a triplet (Parameter, Relation, Finding) to encode the information typically conveyed in a result sentence. There are, however, other

[1]See Schulze-Kremer, Smith, and Kumara (2004) for a description of the shortcomings of the UMLS semantic network.

types of results conveyed in clinical studies. In particular, authors often compare different groups of population with respect to an outcome. For instance, in the following sentence two groups of diabetes patients undergoing two different surgeries are compared: "After adjustments, patients with diabetes undergoing initial PCI were 49% more likely to die during follow-up compared with patients undergoing initial bypass surgery (HR=1.49; CI 95%: 1.02 to 2.17; P=0.037)." TAS does not have any mechanism to represent the comparison reported in this sentence. Although these types of sentences occur often in clinical studies, we chose to ignore them for two reasons. The first reason concerns the difficulty of identify the degree of similarity between the information conveyed in two such sentences. We need this similarity measure to decide whether to merge or at least present together two comparison sentences. It is not clear how to assess the similarity between two comparisons. Second, even if we have a pairwise similarity measure to process comparison sentences, it is still a research question to determine how to fuse the information conveyed by a set of similar comparison sentences. Since we aim to produce a re-generated summary, we need to be able to compare and merge pieces of content from multiple sources. Furthermore, because our summarizer's role is to provide an access point into the input articles, it is reasonable not to select every relevant result in the input studies. Accordingly, we chose a conservative approach and ignored comparisons rather than attempting to extract as many results as possible.

Evaluation in the medical domain. We provide in this thesis intrinsic evaluation for individual modules of the summarizer. In some cases, however, the only way to evaluate a module is by conducting a study with human judges. Because

the summarizer operates over technical medical texts, the judges must be medically knowledgeable. To evaluate our content selection module, we got a medical expert to judge our output, but ideally we would have collected several gold standards from several experts, which would have allowed us to identify variations among judges. For the ordering module, we would have needed several judges to look at several orderings per summary, as we did for the evaluation of the ordering strategy for news summaries. Given the difficulty of recruiting enough physicians for evaluation purposes, we preferred to save them for the overall evaluation of the summarizer.

Personalization and the summarizer's architecture. In our current design, the user's interests are taken into account at the content selection stage but not at successive stages. While it seems obvious that content is highly affected by the user's interests, there is no theoretical reason not to incorporate personalization at the other stages of the summarizer. For instance, the organization of the summary content could be influenced by the input patient's characteristics. In our ordering algorithm, the query terms, which also represent the user's interests, participate in the computation of the ordering priority for blocks of information. This is, however, only a first step towards incorporating the user's interest in the organization stage.

Similarly, the level of expertise of the user is taken into account at the realization stage only. We do not have a mechanism to incorporate such knowledge into other stages of the summarizer. For instance, arguably, knowing whether or not a piece of information is too technical to be understood by the user could prove useful at the content selection stage.

Rewriting strategies. Our strategy to supplement technical terms with a definition could benefit from substituting them with more comprehensible synonyms instead. However, there are no appropriate synonyms for most technical terms. Moreover, previous work on paraphrasing acquisition has shown that synonymy is not always the best lexical relation to find an appropriate paraphrase for a given lexical item (Barzilay and McKeown, 2001). In some cases, for instance, it could make more sense to use a hypernym of a technical term for a lay reader rather than a synonym. This might come at a loss of information from a medical standpoint, but this is inherent to the task of simplification. Our work on sentence alignment is a first step towards acquiring appropriate paraphrases from corpora, but after that much research is still needed to identify which lay phrases are valid alternatives for technical ones.

Finally, while we showed evidence that technical and lay languages differ at the sentence and rhetorical levels, our approach does not attempt to adapt the syntactic structure of a given sentence, nor does it try to reorganize the different sentences in a given text to fit the structure of lay texts better.

8.3 Future Work

In this thesis, we have examined only one of the many phenomena that occur when adapting a technical text into a simpler version. We believe that the task of transforming such texts, and more generally the task of text-to-text generation is an exciting field. Here we suggest directions for future work in this area.

Different levels of expertise. When we consider the task of rewriting to adapt to the level of expertise of a reader, it might be too naive to assume a binary classification of users, as in the technical/lay paradigm. One exciting research direction is to identify degrees of expertise and adjust which rules to apply and how to apply them accordingly until the text reaches an appropriate degree of technicality. Readability measures have been extensively investigated to determine the degree of expertise required to understand a given text, but none have been satisfying so far. Is it possible to learn such readability measures from text? And finally, can this measure be used as a yardstick for selecting rewriting rules during re-generation?

Transformations at the sentence level. The two following sentences taken from our comparable corpus convey the same information.

> "The age-adjusted RRs for heart disease corresponding to intakes of <0.5 drinks/day were 0.76 (95% confidence interval: [CI]: 0.52 to 1.12) as compared with nondrinkers (p for TREND = 0.06)."
>
> "Compared with nondrinkers, men who consumed half a drink or less per day cut their heart disease risk by 24%."

Without the benefit of domain knowledge, it is not easy to determine that these two sentences do, in fact, paraphrase each other: parenthetical information can be safely dismissed, and "a risk ratio of 0.76" means the same as "cut their risk by 24%." At the same time, such paraphrases should be learned from examples, as there are many such types of sentence-level paraphrases.

Transformations at the textual level. One interesting research question is what types of transformations, beyond rewriting each sentence in isolation, are needed at the discourse level to ensure a better quality of re-generated text. Increasing the cohesion of a generated text can be achieved in several ways. Locally, cohesion links can be added as a post-processing stage, by making use of anaphoras for instance. Globally, reordering the information to fit the style of lay texts better can enhance the understanding of the rewritten text for the readers. An exciting direction for future work is investigating how to map one document structure typical of one style, such as technical, to another document structure typical of the target style. A mapping mechanism would first require identifying such document structures. In traditional generation systems, schemas were identified through manual analysis of texts. More recently, there has been work on learning document structures automatically in a knowledge-lean fashion. Depending on how the document structure for each style was acquired, a mapping mechanism will accordingly depend upon a more or less knowledge-lean approach.

Domain adaptation. As it is not always possible to learn rewriting rules in a new domain (because for instance of lack of training data), transferring rules from another domain is an attractive approach. One challenging question, therefore, is whether all rewriting rules are domain-dependent, or whether there are some universal rewriting rules that carry across domains. While sentence-level paraphrases are most likely specific to a given domain, what about some of the rewriting rules at the syntactic or lexical levels for instance?

Interactions among the different types of transformations. As the rules for rewriting affect different levels of a text (lexical, syntactic, sentential, and discourse), one interesting question is to what extent each type of transformation affects the others. Research to identify which architecture provides better quality texts overall is needed. There are many research questions: Is it possible to find an appropriate order in which to apply the rewriting rules? Or is it possible to implement a feedback mechanism to allow for flexible rule selection? More generally, is there such a thing as a consensus architecture for text-to-text rewriting? These are some of the many interesting questions involved in designing a text-to-text generation system.

8.4 Sentence Alignment as a First Step towards Simplification

If we consider the task of transforming a technical sentence into a simpler version a translation task, an attractive strategy is to learn rewriting rules or translation rules the way machine translation (MT) systems do. There are, however, at least two important challenges to applying MT techniques to the simplifcation task: (1) Any MT technique relies on a corpus of aligned sentences. Traditionally such sentences pairs are identified automatically from a parallel corpus. In our case, there is no parallel corpus of technical/lay texts, as simplification is not only a function of the language used but also the content delivered to the reader. It is still possible to find instances of aligned sentences, but these will be mined from comparable corpora. Sentence alignment techniques designed for parallel corpora do not carry well to

comparable corpora. Thus, a new algorithm is needed to identify sentence pairs in comparable corpora. (2) It is difficult to find large amount of sentence pairs which convey similar information but are targeted at different audiences like in traditional MT. Thus, a technique developed for the simplification task will have to learn from a small number of examples.

In this section we present an algorithm we designed to align sentences in a monolingual corpus. This addresses the first challenge described above. We first go through the challenges entailed in aligning sentences in a comparable corpus and highlight the contributions of our approach. We provide an overview of previous work on monolingual sentence alignment and describe our alignment algorithm. Finally we report on the data on which we developed and tested our algorithm and the evaluation of our method.[2]

This work was originally tested on a different domain from the medical one of our framework. While we applied this method to medical data, namely to a subset of the ReutersHealth stories and their corresponding technical clinical studies, we did not achieve satisfactory results. A discussion is provided at the end of this section.

8.4.1 Challenges and Contributions

In MT, the task of sentence alignment was extensively studied for parallel corpora and can be considered solved for close languages, such as English and French. Melamed (1999) reports an error rate of 1.8% on the standard Hansard corpus. A typical sentence alignment algorithm can be roughly described as a two-step

[2]This section is based on the EMNLP paper (Barzilay and Elhadad, 2003).

process: (1) for each sentence pair compute a local similarity value, independently of the other sentences; (2) find an overall sequence of mapped sentences, using both the local similarity values and additional features.

In the case of monolingual corpora, step (2) might seem unnecessary. Since the texts share the same language, it would be enough to choose for local similarity a function based on lexical cues only and select sentence pairs with high lexical similarity. Even a simple lexical function (e.g., one that counts word overlap) could produce an accurate alignment. After all, two sentences which share most of their words are likely to paraphrase each other. The problem is that there are many sentences that convey the same information but have little surface resemblance. As a result, simple word counts cannot distinguish the matching pair (A) in Figure 8.1 from the unrelated pair (B). An accurate local similarity measure would have to account for many complex paraphrasing phenomena. But, frustratingly, to learn such transformations automatically, sentences must first be aligned. A simple, weak lexical similarity function alone is not sufficient.

(A)	· Petersburg served as the capital of Russia for 200 years. · For two centuries Petersburg was the capital of the Russian Empire.
(B)	· The city is also the country's leading port and center of commerce. · And yet, as with so much of the city, the port facilities are old and inefficient.

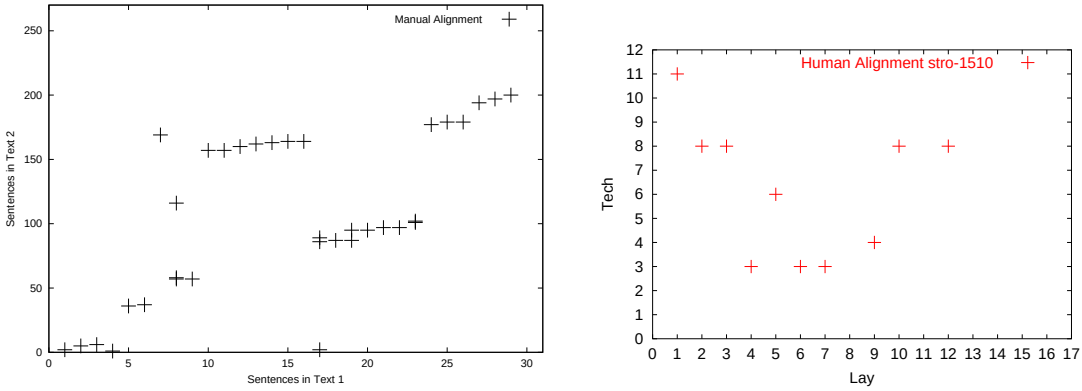
Figure 8.1: Sentence pairs from our comparable corpus with two content words in common. (A) is a matching pair, (B) is not.

In MT, a weak similarity function is compensated for by searching for a globally optimal alignment, using dynamic programming or taking advantage of the geometric/positional or contextual properties of the text pair (Gale and Church,

1991; Shemtov, 1993; Melamed, 1999). But these techniques operate on the assumptions that there are limited insertions and deletions between the texts and that the order of the information is roughly preserved from one text to another.

Texts from comparable corpora, as opposed to parallel corpora, contain a great deal of "noise." Figure 8.2 shows the manual alignment determined by a judge of two text pairs in two different corpora we experimented with: Britannica (Figure 8.4.1) and Reuters (Figure 8.4.1). In the long Britannica texts, only a small fraction of the sentences got aligned (35 out of 31×270 sentence pairs), which illustrates that there is no complete information overlap. For instance, consider two texts written for different audiences: while both convey the same information, one may contain technical details and the other, background information. Another distinguishing characteristic of comparable corpora is that the order in which the information is presented can differ greatly from one text to another. Analysis of comparable texts in different domains (Paris, 1993; Barzilay, Elhadad, and McKeown, 2002) showed that there is wide variability in the order in which the same information can be presented. This is illustrated in both plots: the dots clearly indicate that one cannot expect the order in which the information is presented to be preserved from one type of text to the other. As a result of these two factors – noise and order –, we cannot take advantage of the positional features used in MT to find a global mapping.

We investigate a novel approach informed by text structure for sentence alignment. Our method emphasizes the search for an overall alignment, while relying on a simple local similarity function. We incorporate context into the search process in two complementary ways: (1) we map large text fragments using hy-



(a) Britannica (b) Reuters

Figure 8.2: Manual alignments for two text pairs in the Britannica and Reuters comparable corpora. A dot on (x,y) indicates that sentences x and y match.

potheses learned in a supervised fashion and (2) we further refine the match through local alignment within mapping fragments to find sentence pairs. When the documents in the collection belong to the same domain and genre, the fragment mapping takes advantage of the topical structure of the texts. Our experiments show that our overall approach identifies even pairs with low lexical similarity. We also found that a fully unsupervised method using a minimalist representation of contextual information, *viz.*, paragraph-level lexical similarity, outperforms existing methods based on complex local similarity functions.

8.4.2 Related Work

Most of the work in monolingual corpus alignment is in the context of summarization. In single document summarization, alignment between full documents and summaries written by humans is used to learn rules for text compression. Marcu (1999) computes sentence similarity using a cosine-based metric. This metric was later used for identifying plagiarism across documents, as well as analyzing word-

reuse from one text to another (Clough et al., 2002). Marcu's method is as follows: a document is first parsed to get its discourse structure. Satelite clauses are pruned off the search. Remaining clauses are aligned against summary sentences using the cosine metric. The alignment is thus at the clause level.

Jing's tool Decomposition (2002) identifies phrases that were cut and pasted together using a Hidden Markov Model with features incorporating word identity and positioning within sentences, thereby providing an alignment of the document and its summary at the phrase level. The transition probabilities of the HMM were assigned manually based on general heuristics rules. Both of these methods – Marcu's and Jing's – construct an alignment by looking at text units (clauses or phrases) one at a time, independently of the decisions made about other text units. Because summaries often reuse original document text to a large extent, these methods achieve good results.

In the context of multidocument summarization, SimFinder (Hatzivassiloglou et al., 2001) identifies sentences that convey similar information across input documents to select the summary content. Even though the input documents are about the same subject, they exhibit a great deal of lexical variability. To address this issue, SimFinder employs a complex similarity function, combining features that extend beyond a simple word count and include noun phrase, proper noun, and WordNet sense overlap. Since many documents are processed in parallel, clustering is used to combine pairwise alignments. In contrast to our approach, SimFinder does not take the context around sentences into account.

Comparable bilingual corpora have primarily been used to build bilingual lexical resources (Fung and Yee, 1998). More recently, Fung and Cheung (2004)

proposed an algorithm to simultaneously mine sentence alignments and lexicon in a comparable bilingual corpus.

8.4.3 Methods

Since computing an accurate domain-specific local similarity measure is a daunting task, we propose an algorithm that shifts the burden to the search for a valid overall mapping by incorporating information about the context in which the sentences appear. The key features of our approach are: (1) Pruning the search space at the paragraph level and (2) performing local alignment within paragraph pairs compensate for a weak similarity measure.

Given a comparable corpus consisting of two collections and a training set of manually aligned text pairs from the corpus, the algorithm follows four main steps. Steps 1 and 2 take place at training time. Steps 3 and 4 are carried out when a new text pair ($Text_1$, $Text_2$) is to be aligned.

1. **Topical structure induction**: by analyzing multiple instances of paragraphs within the texts of each collection, the topics characteristic of the collections are identified through clustering. Each paragraph in the training set gets assigned the topic it verbalizes.

2. **Learning of structural mapping rules**: using the training set, rules for mapping paragraphs are learned in a supervised fashion (Section 8.4.3). Features for the learning include the paragraph topics and their lexical similarity.

3. **Macro alignment**: given a new unseen pair ($Text_1$, $Text_2$), each paragraph is automatically assigned its topic. Paragraphs are mapped following the

learned rules.

4. **Micro alignment**: for each mapped paragraph pair, a local alignment is computed at the sentence level. The final alignment for the text pair is the union of all the aligned sentence pairs.

Off-Line Processing

Given two sentences with moderate lexical similarity, we may not have enough evidence to decide accurately whether they should be aligned. Looking at the broader context they appear in can provide additional insight: if the *types* of information expressed in the contexts are similar, then the *specific* information expressed in the sentences is more likely to be the same. On the other hand, if the types of information in the two contexts are unrelated, chances are that the sentences should not be aligned. In our implementation, context is represented by the paragraphs to which the sentences belong.[3] Our goal in this phase is to learn rules for determining whether two paragraphs are likely to contain sentences that should be aligned, or whether, on the contrary, two paragraphs are unrelated and, therefore, should not be considered for further processing.

A potentially fruitful way to do so is to take advantage of the topical structure of texts. In a given domain and genre, while the texts relate different subjects, they all use a limited set of topics to convey information; these topics are also known as the Domain Communication Knowledge (Kittredge, Korelsky, and Rambow, 1991). For instance, most texts describing diseases will have topics such as "symptoms" or

[3]Texts without adequate paragraph marking could be segmented using tools such as TextTiling (Hearst, 1994).

"treatment."[4] If the task is to align a disease description written for physicians and a text describing the same disease for lay people, it is most likely that sentences within the topic "symptoms" in the expert version will map to sentences describing the symptoms in the lay version rather than those describing treatment options. If we can automatically identify the topic each paragraph conveys, we can decide more accurately whether two paragraphs are related and should be mapped for further processing.

In the field of text generation, methods for representing the semantic structure of texts have been investigated through text schemata (McKeown, 1985) or rhetorical structures (Mann and Thompson, 1988). In our framework, we want to identify the different topics of the text, but we are not concerned with the relations holding between them or the order in which they typically appear. We propose to identify the topics typical to each collection in the comparable corpus by using clustering, such that each cluster represents a topic in the collection.

The process of learning paragraph mapping rules is accomplished in two stages: first, we identify the topics of each collection, $Corpus_1$ and $Corpus_2$, and label each paragraph with its specific topic. Second, using a training set of manually aligned text pairs, we learn rules for mapping paragraphs from $Corpus_1$ to $Corpus_2$. Two paragraphs are considered mapped if they are likely to contain sentences that should be aligned.

Vertical Paragraph Clustering We perform a clustering at the paragraph level for each collection. We call this stage Vertical Clustering because all the paragraphs

[4]We use the term topic differently than it is commonly used in the topic detection task– there, a "topic" would designate which disease is described.

Lisbon has a mild and equable climate, with a mean annual temperature of 63 degree F (17 degree C). The proximity of the Atlantic and the frequency of sea fogs keep the atmosphere humid, and summers can be somewhat oppressive, although the city has been esteemed as a winter health resort since the 18th century. Average annual rainfall is 26.6 inches (666 millimetres).
Jakarta is a tropical, humid city, with annual temperatures ranging between the extremes of 75 and 93 degree F (24 and 34 degree C) and a relative humidity between 75 and 85 percent. The average mean temperatures are 79 degree F (26 degree C) in January and 82 degree F (28 degree C) in October. The annual rainfall is more than 67 inches (1,700 mm). Temperatures are often modified by sea winds. Jakarta, like any other large city, also has its share of air and noise pollution.

Figure 8.3: Two automatically clustered paragraphs in the same collection (without date, number, and name substitution).

of all the documents in $Corpus_1$ get clustered, independently of $Corpus_2$; the same goes for the paragraphs in $Corpus_2$. At this stage, we are only interested in identifying the topics of the texts in each collection, each cluster representing a topic.

We apply a hierarchical complete-link clustering. Similarity is a simple cosine measure based on the word overlap of the paragraphs, ignoring function words. Since we want to group together paragraphs that convey the same type of information across the documents in the same collection, we replace all the text-specific attributes, such as proper names, dates and numbers, by generic tags.[5] This way, we ensure that two paragraphs are clustered not because they relate the same specific information, but rather, because they convey the same type of information (an example of two automatically clustered paragraphs is shown in Figure 8.3). The number of clusters for each collection is a parameter tuned on our training set (see Section 8.4.3).

[5]We crudely consider any words with a capital letter a proper name, except for each sentence's first word.

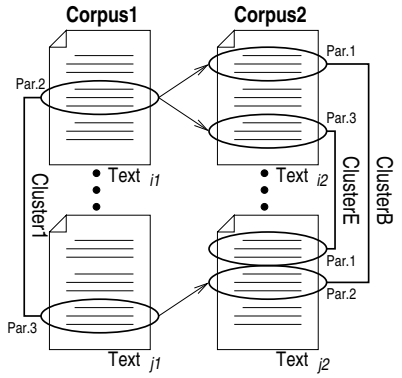


Figure 8.4: The training set for the paragraph mapping step. An arrow between two paragraphs indicates they contain at least one aligned sentence pair.

Horizontal Paragraph Mapping Once the different topics, or clusters, are identified inside each collection, we can use this information to learn rules for paragraph mapping (Horizontal Mapping between texts from $Corpus_1$ and texts from $Corpus_2$). Using a training set of text pairs, manually aligned at the sentence level, we consider two paragraphs to map each other if they contain at least one aligned sentence pair (see Figure 8.4).

Our problem can be framed as a classification task: given training instances of paragraph pairs (P, Q) from a text pair, classify them as mapping or not. The features for the classification are the lexical similarity of P and Q, the cluster number of P, and the cluster number of Q. Here, similarity is again a simple cosine measure based on the word overlap of the two paragraphs.[6] These features are weak indicators by themselves. Consequently, we use the publicly-available classification tool BoosTexter (Singer and Schapire, 1998) to combine them accurately.[7]

[6]At this stage, we want to match on text-specific information, unlike in the Vertical Clustering. We therefore use the original text, without any substitution, to compute the similarity.

[7]Because BoosTexter cannot form conjunctive hypotheses, we add a feature which encodes the combination of two cluster numbers.

Macro Alignment: Find Candidate Paragraph(s)

At this stage, the clustering and training are completed. Given a new unseen text pair ($Text_1$, $Text_2$), the goal is to find a sentence alignment between them. Two sentences with very high lexical similarity are likely to be aligned. We allow such pairs in the alignment independently of their context. This step allows us to catch the "easy" paraphrases. We focus next on how our algorithm identifies the less obvious matching sentence pairs.

For each paragraph in each text, we identify the cluster in its collection it is the closest to. Similarity between the paragraph and each cluster is computed the same way as in the Vertical Clustering step. We then apply mapping classification to find the mapping paragraphs in the text pair (see Figure 8.5).

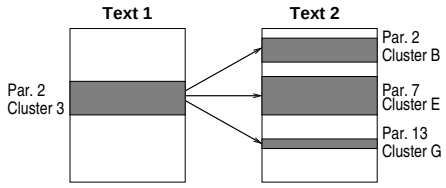


Figure 8.5: Macro Alignment: a paragraph in $Text_1$ and its mapped candidate paragraphs in $Text_2$.

Micro Alignment: Find Sentence Pair(s)

Once the paragraph pairs are identified in ($Text_1$, $Text_2$), we want to find, for each paragraph pair, the (possibly empty) subsets of sentence pairs which constitute a good alignment. Context is used in the following way: given two sentences with moderate similarity, their proximity to sentence pairs with high similarity can help us decide whether to align them or not.

To combine the lexical similarity (again using cosine measure) and the proximity feature, we compute local alignments on each paragraph pair, using dynamic programming. The local alignment we construct fits the characteristics of the data we are considering. In particular, we adapt it to our framework to allow many-to-many alignments and some flips of order among aligned sentences. Given sentences i and j, their local similarity $sim(i, j)$ is computed as follows, based on their cosine similarity and the mismatch penalty:

$$sim(i, j) = cos(i, j) - mismatch\_penalty$$

The weight $s(i, j)$ of the optimal alignment between the two sentences is computed as follows:

$$s(i, j) = max \begin{cases} s(i, j-1) - skip\_penalty \\ s(i-1, j) - skip\_penalty \\ s(i-1, j-1) + sim(i, j) \\ s(i-1, j-2) + sim(i, j) + sim(i, j-1) \\ s(i-2, j-1) + sim(i, j) + sim(i-1, j) \\ s(i-2, j-2) + sim(i, j-1) + sim(i-1, j) \end{cases}$$

The mismatch penalty penalizes sentence pairs with very low similarity measure, while the skip penalty prevents *only* sentence pairs with high similarity from getting aligned.

Evaluation Design

The Data We compiled a comparable corpus to test our method consisting of encyclopedia articles written for different audiences. The Britannica corpus contains texts from Encyclopedia Britannica and Britannica Elementary. In contrast to the

	Sentences			Paragraphs		
	Min	Max	Avg	Min	Max	Avg
Britannica Comp. Train	87	313	180	19	59	37
Britannica Comp. Test	138	308	200	32	63	43
Britannica Elem. Train	34	64	47	8	12	10
Britannica Elem. Test	27	75	45	6	16	10

Table 8.1: Statistics for the training and testing sets for the comprehensive/lay versions of Britannica.

long (up to 15-page) detailed articles of the Encyclopedia Britannica, Britannica Elementary contains one- to two-page entries targeted towards children. The elementary version generally contains a subset of the information presented in the comprehensive version, but there are numerous cases when the elementary entry contains additional or more up-to-date pieces of information.[8] The two collections together exhibit many instances of complex rewriting. We collected 103 pairs of comprehensive/elementary city descriptions. We set aside a testing set of 11 text pairs. The rest (92 pairs) was used for the Vertical Clustering. Nine text pairs were used for training (see Table 8.1 for statistics).

Human Annotation. Each text pair in the training and testing sets was annotated by two annotators.[9] In our guidelines to them, we defined two sentences as aligned if they contain at least one clause that expresses the same information. We allowed many-to-many alignments. Several labeled examples were provided to supply further guidance. On average, each annotator spent 50 minutes per text pair (shortest time was 35 minutes, longest was two hours). While the annotators agreed

[8]Britannica Elementary is a new feature of the encyclopedia, not all entries in the original Britannica have been fully updated.

[9]All the annotators were native speakers of English.

Range	Training	Testing
0%–40%	149 (46.6%)	127 (45.2%)
40%–70%	103 (32.2%)	96 (34.2%)
70%–100%	68 (21.2%)	58 (20.6%)

Table 8.2: Distribution of manually aligned sentence pairs among different similarity ranges in the Britannica corpus.

for most of the sentence pairs they identified, there were some cases of disagreement. Alignment is a tedious task, and sentence pairs can easily be missed even by a careful human annotator. For each text pair, a third annotator went through contested sentence pairs, deciding on a case-by-case basis whether to include it in the alignment.

Overall, 320 sentence pairs were aligned in the training set and 281 in the testing set. The other sentence pairs which were not aligned served as negative examples, yielding a total of 4192 training instances and 3884 testing instances. Our corpus is available at `http://www.cs.columbia.edu/~noemie/alignment`.

As a confirmation that there is no order preservation in comparable corpora, there were up to nine order shifts in each of the annotated text pairs. In addition, our analysis confirms that aligned sentences (as identified by humans) do not share many content words. Table 8.2 shows that a large fraction of manually aligned sentence pairs have low lexical similarity (46.6% of aligned pairs in the Britannica training set have up to 40% similarity, while only 21.2% have more than 70% similarity). Similarity is measured here by the number of words in common, normalized by the number of types in the shorter sentence.

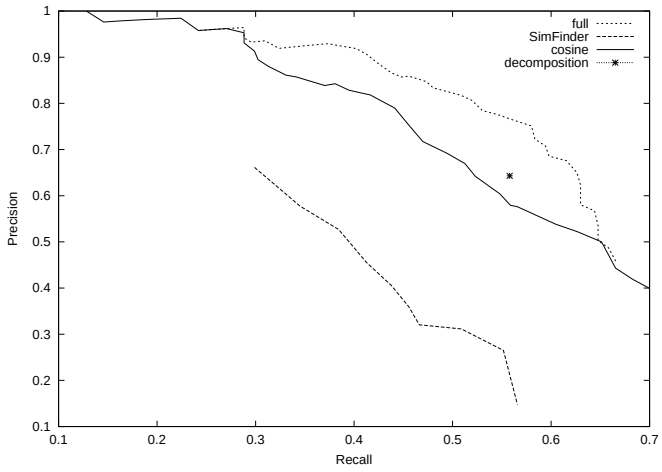


Figure 8.6: Precision/recall for SimFinder, Cosine, Decomposition, and our full method.

Parameter Tuning. We tuned all the parameters on our training set, obtaining the following values: For the Britannica corpus; the skip penalty is 0.001, and the cosine threshold for selecting pairs with high lexical similarity is 0.5. BoosTexter was trained in 200 iterations. To find the optimal number of clusters for each collection, Vertical Clustering was performed with different numbers of clusters, ranging from 10 to 40; we selected the alternatives with the best performance on the training set: 20 for both collections.

8.4.4 Evaluation

We carried out two types of evaluations: we compared our method with other systems developed for the same task and we analyzed the impact of individual components on the performance of our method.

System	Precision
SimFinder	24%
Cosine	57.9%
Decomposition	64.3%
Without Topic Mapping	71.6%
Without Local Alignment	73.3%
Full Method	**76.9%**
Full Method (with ideal clusters)	80.9%

Table 8.3: Precision at 55.8% recall.

Comparison with Other Systems

An obvious choice for a baseline in this task is the following: any two sentences are considered aligned if their cosine similarity exceeds a certain threshold. We also compare our algorithm with two state-of-the-art systems, SimFinder (Hatzivassiloglou et al., 2001) and Decomposition (Jing, 2002).[10] Figure 8.6 shows precision/recall curves for the different systems. For our full system, we obtain different values of recall keeping constant the skip penalty and the cosine threshold and varying the value of the mismatch penalty from 0 to 0.45.[11] This setup results in recall values in a 25%–65% range. The curve for SimFinder was obtained by running SimFinder with different similarity thresholds, ranging from 0.1 to 0.95. In the case of Decomposition, there are several hard-coded parameters which are not trainable. As a result, we were able to obtain results for Decomposition only at a 55.8% recall level. Table 8.3 reports precision values at this level of recall.

Our full method outperforms both the baseline ("*Cosine*") and the more

[10] Jing (2002) reports that Decomposition outperforms the algorithm of Marcu (1999); we, therefore, did not compare our method against Marcu's system.

[11] Varying the mismatch penalty is a natural choice: varying the skip penalty produces a narrow range of recall values, while the cosine threshold controls only a small portion of the sentence pairs that can be identified (the ones with high similarity).

(*)	Gradually the German culture and language became more widespread in the city.
	Capping Prague's rebirth, it was designated a European City of Culture in 2000.
	Prague is a centuries-old city with a wealth of historic landmarks.
	The physical attractions and landmarks of Prague are many.

Figure 8.7: Aligned pairs, (*) denotes an incorrect alignment.

complex systems ("*SimFinder*" and "*Decomposition*"). Interestingly, methods that use simple local similarity functions significantly outperform SimFinder (SimFinder was trained on newswire texts; we did not have access to SimFinder's training component for retraining on our corpus). This confirms our hypothesis that while it is an appealing idea, putting all one's eggs in the basket of a sophisticated local similarity measure to achieve good performance may be too hard a task. The simple cosine baseline is competitive with the Hidden Markov Model of Decomposition (Decomposition was specifically developed to identify sentence pairs with cut-and-paste transformations, not all possible paraphrase pairs). This suggests that when looking for an alignment, Cosine is a good, yet simple, starting local similarity measure. Adding on top of it an explicit search mechanism relying on the context surrounding the sentences, as in our method, results in a performance improvement of 19% at 55.8% recall. Figure 8.7 shows examples of pairs identified by our method.

Analysis

Impact of Horizontal Paragraph Mapping. We hypothesize that exploiting the regularity in mapping between semantic units, such as topics, improves the alignment. We compared the performance of our full method with a variation that does not take any topical information into account. For the paragraph mapping,

we replaced the learned rules by a single rule based on lexical similarity: two paragraphs are mapped if their cosine measure is above a pre-specified threshold.[12] This new mapping is a good point of comparison because it does not rely on any knowledge inferred from the other texts in the corpus. The results confirm our hypothesis: learning paragraph mapping based on topical structures improves the performance (see "*Without Topic Mapping*" vs. "*Full Method*", Table 8.3).

This experiment also shows that representing context, even simply using only the paragraphs and their lexical similarity, achieves higher performance than methods based on more complex local similarity functions. It is an important finding, because this simplified method can be used when topic structure cannot be derived (e.g., in heterogeneous collections) or when no training data is available, since it is unsupervised.

Impact of Cluster Quality. Our method uses clustering to identify the different topics of each collection. It is important to know how sensitive our overall algorithm is to the quality of the identified clusters. Fortunately, in our corpus, some of the texts contain section headings (e.g., "Climate" or "Demography"). Even though our method ignores this piece of information, we used it to derive manually "ideal" clusters.[13] We obtained eight clusters for the elementary version and 11 for the comprehensive one. When feeding these ideal clusters instead of the automatically identified ones to the learning module for paragraph mapping, we achieve 4% improvement in precision (at 55.8% recall). We interpret this as a sign that the

[12]The threshold was tuned on our training data.

[13]The process was performed manually because the sections are different from one text to another, both in names and levels of detail, and because we needed to infer clusters for the paragraphs that did not have section headings.

Range	Full		Dec.		Cos.	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
0%–40%	50%	25%	34%	28%	23%	15%
40%–70%	85%	73%	82%	74%	66%	86%
70%–100%	95%	95%	93%	88%	90%	95%

Table 8.4: Precision and recall for different ranges of lexical similarity for Decomposition, Cosine, and our full method.

algorithm handles imperfect, automatically induced clusters fairly well.

Impact of Local Alignment. Our hypothesis for computing local alignments between pairs of mapped paragraphs is that our approach allows us to identify additional matching sentence pairs: if two sentences paraphrase each other but have a low cosine measure, looking at the sentence pairs around them may increase their chances of getting selected.

We compared our full method with a version of our algorithm that does not perform local alignment ("*Without Local Alignment*"). Instead, it simply selects sentence pairs inside the mapped paragraphs based on their cosine measure. This incomplete version of the algorithm achieves 73.3% precision (at 55.8% recall), 3% lower than the full method, validating our hypothesis.

Impact of Lexical Similarity. We investigated how the performance of our method depends on the lexical similarity of the input sentences. Table 8.4 shows precision and recall for our method and others at three sentence-similarity ranges based on word overlap counts (at the overall recall of 55.8%). Our method outperforms the Cosine baseline and Decomposition on all similarity ranges.

8.4.5 In the Medical Domain

We applied our algorithm to parallel texts from a subset of the ReutersHealth corpus and their corresponding clinical studies. While our method performs better than the Cosine baseline, and the full method performs better than the two modified versions (without topic mapping and without local alignment), it does not yield results satisfactory enough to collect accurate instances of technical/simplified sentence pairs. We identified the two following challenges that need to be addressed.

Topical structure of ReutersHealth texts. It is obvious that stories in the ReutersHealth corpus have a strong topical structure. ReutersHealth stories are composed of background information, methods, results, and interviews with the authors of the original study. This is promising because it means that we can take advantage of the structure to align the sentences between ReutersHealth stories and technical clinical studies, which also exhibits a strong topical structure. Because in the technical studies the language typically used for methods is quite different from the language used for results and so forth, we can apply the same method as for the Britannica corpora to identify the different topics, namely using a similarity measure based on word overlap. However, the language used in ReutersHealth stories is not formulaic in the same way. Topic are insufficiently differentiated from a purely lexical standpoint.

We also experimented with a supervised approach. Selecting a training set of ReutersHealth stories, we manually annotated 600 paragraphs as methods, background, results, conclusions, quote or other. We ran the classifier Boostexter with features of a paragraph, its text and the ngrams it contains. The classifier achieves

acceptable recognition of results paragraphs (69% precision and 64% recall) and quotes (68% precision and 64% recall), but does not achieve good results for other types of discourse units (methods are recognized with 48% precision and 47% recall, for instance, while background paragraphs are recognized with 70% recall but 50% precision). Because the identification of the topical structure is the first step in the alignment process, these results are not satisfactory enough to allow the use of automatically identified topics for this domain. Automatically identifying a topical structure for a set of texts in the same domain and genre, is a difficult task, especially when the domain is not highly formulaic.

Lexical similarity between ReutersHealth texts and clinical studies. In addition, in the Britannica corpus, we found that the adult and children texts exhibit lexical differences sufficient to render a simple measure like the Cosine baseline inefficient at identifying sentence pairs that convey similar information. In the medical domain, the gap in wording between the clinical studies and their lay counterpart is even wider. Thus, there is very little lexical overlap between matching sentences. As a matter of fact, when we compared the word overlap between a set of manually matched sentence pairs and a set of sentences paired randomly, there was no statistical difference between the two. We experimented with a few variants of the lexical similarity measure, one of them being to replace each term with its CUI to abstract over the lexical differences. But none of them helped overall.

A promising avenue for research is to learn a similarity measure from examples of matching and non-matching sentences. However, this requires many examples of matching sentences. Because manual sentence alignment is a time-consuming task, we did not investigate this approach any further.

References

Abney, Steven. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344.

Abrams, M. H. 1953. *The Mirror and the Lamp: Romantic Theory and the Critical Tradition*. Oxford University Press.

Bangalore, Srinivas, Owen Rambow, and Steve Whittaker. 2000. Evaluation metrics for generation. In *Proceedings of the International Natural Language Generation Conference (INLG'00)*, pages 1–8.

Barzilay, Regina and Michael Elhadad. 1998. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization ACL Workshop*.

Barzilay, Regina and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'03)*, pages 25–32.

Barzilay, Regina, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, Regina and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'05)*, pages 141–148.

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised

approach using multiple-sequence alignment. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 16–23.

Barzilay, Regina and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'04)*, pages 113–120.

Barzilay, Regina and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'01)*.

Barzilay, Regina and Kathleen McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297 – 327.

Bateman, John and Cecile Paris. 1989. Phrasing a text in terms the user can understand. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1511–1517.

Becher, Margit, Brigitte Endres-Niggemeyer, and Gerrit Fichtner. 2002. Scenario forms for web information seeking and summarizing in bone marrow transplantation. In *Proceedings of the COLING Workshop on Multilingual Summarization and Question Answering*.

Binsted, Kim, Alison Cawsey, and Ray Jones. 1995. Generating personalised patient information using the medical record. In *Proceedings of Artificial Intelligence in Medicine Europe*.

Bouayad-Agha, Nadjet, Richard Power, and Donia Scott. 2000. Can text structure be incompatible with rhetorical structure? In *Proceedings of the International Natural Language Generation Conference (INLG'00)*, pages 194–200.

Carenini, Giuseppe, Vibhu Mittal, and Johanna Moore. 1994. Generating patient specific interactive explanations. In *Symposium on Computer Applications in Medical Care*.

Carroll, John, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL'99)*.

Cawsey, Alison, Ray Jones, and Janne Pearson. 2000. The evaluation of a personalised information system for patients with cancer. *User Modeling and User-Adapted Interaction*, 10(1):47–72.

Chandrasekar, Raman and Srinivas Bangalore. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183–190.

Clegg, Andrew and Adrian Shepherd. 2005. Evaluating and integrating treebank parsers on a biomedical corpus. In *Proceedings of the ACL Workshop on Software 2005*.

Clough, Paul, Robert Gaizauskas, Scott Piao, and Yorick Wilks. 2002. METER: MEasuring TExt Reuse. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'02)*.

Coltheart, M. 1981. The MRC psycholinguistic database. *Quarterly Journal of Experimental Psychology*, 33(A):497–505.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Dale, Robert. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press.

DeJong, G., 1982. *Strategies for Natural Language Processing*, chapter An Overview of the FRUMP System, pages 149–176. Lawrence Erlbaum Associates.

Dras, Mark. 1999. *Tree Adjoining Grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Macquarie University.

Duboue, Pablo and Kathleen McKeown. 2001. Empirically estimating order constraints for content planning in generation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL'01)*, pages 172–179.

DUC. 2002-2005. Proceedings of the document understanding conference.

Ebadollahi, Shahram, Shih-Fu Chang, Henry Wu, and Shin Takoma. 2001. Indexing and summarization of echocardiogram videos. In *Scientific Session of the American College of Cardiology*.

Eco, Umberto. 1979. *The Role of the Reader : Explorations in the Semiotics of Texts*. Indiana University Press.

Edmundson, H. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Edwards, Kari and Edward Smith. 1996. A disconfirmation bias in the evaluation of arguments. *Journal of Personality and Social Psychology*, 71:5–24.

Elhadad, Noemie, Kathleen McKeown, David Kaufman, and Desmond Jordan. 2005. Facilitating physicians' access to information via tailored text summarization. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'05)*.

Filatova, Elena and Vasilieos Hatzivassiloglou. 2004. Event-based extractive summarization. In *Proceedings of the ACL Text Summarization Workshop*.

Forster, K., 1976. *New approaches to language mechanisms*, chapter Accessing the mental lexicon, pages 257–276. R. Wales and E. Walker.

Fung, Pascale and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*.

Fung, Pascale and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'98)*.

Gale, William and Kenneth Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'91)*.

Gerrig, Richard, Jennifer Kuczmarski, and Susan Brennan. 1999. Perspective effects on readers' text representations. In *NSF HCI Program. Grantees' Workshop*.

Goldstein, Jade, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 121–128.

Green, Noah, Panagiotis Ipeirotis, and Luis Gravano. 2001. SDLIP + STARTS = SDARTS: A protocol and toolkit for metasearching. In *Proceedings of Joint Conference on Digital Libraries (JCDL'01)*, pages 207–214.

Grefenstette, Gregory. 1998. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working notes of the AAAI Spring Symposium on Intelligent Text Summarization*, pages 111–117.

Halliday, M. and R. Hasan. 1976. *Cohesion in English*. Longman.

Hasan, R., 1984. *Understanding reading comprehension: Cognition, language and the structure of prose*, chapter Coherence and Cohesive Harmony, pages 181–219. International Reading Association.

Hatzivassiloglou, Vasileios, Judith Klavans, Melissa Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. 2001. SimFinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL Workshop on Automatic Summarization*.

Hearst, Marti. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'94)*.

Hovy, Eduard. 1988. Two types of planning in language generation. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'88)*, pages 170–186.

Hovy, Eduard. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1-2):341–386.

Hripcsak, George, James Cimino, and Soumitra Sengupta. 1999. WebCIS: Large scale deployment of a web-based clinical information system. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'99)*.

Inui, Kentaro and Ulf Hermjakob, editors. 2003. *The Second International Workshop on Paraphrasing*. In conjunction with ACL'03.

Iordanskaja, Lidja, Richard Kittredge, and Alain Polguere, 1991. *Natural language generation in artificial intelligence and computational linguistics*, chapter Lexical selection and paraphrase in a meaning-text generation model, pages 293–312. Kluwer.

Jastrzembski, James. 1981. Multiple meanings, number of related meanings, frequency of occurrence, and the lexicon. *Cognitive Psychology*, 13(2):278–305.

Jing, Hongyan. 2002. Using Hidden Markov Modeling to decompose human-written summaries. *Computational Linguistics*, 28(4):527–543.

Jing, Hongyan, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1998. Summarization evaluation methods: Experiments and analysis. In *AAAI Symposium on Intelligent Summarization*, pages 51–59.

Jing, Hongyan and Kathleen McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 178–185.

Kan, Min-Yen. 2003. *Automatic text summarization as applied to information retrieval: Using indicative and informative summaries.* Ph.D. thesis, Columbia University. Chapters 3-4.

Kan, Min-Yen, Judith Klavans, and Kathleen McKeown. 1998. Linear segmentation and segment relevence. In *Proceedings of the International Workshop of Very Large Corpora (WVLC'88).*

Kan, Min-Yen and Kathleen McKeown. 2002. Corpus-trained text generation for summarization. In *Proceedings of the International Natural Language Generation Conference (INLG'02)*, pages 1–8.

Kaufman, David, Vimla Patel, C. Hilliman, P. Morin, J. Pevzner, Weinstock, R. Goland, S Shea, and J. Starren. 2003. Usability in the real world: Assessing medical information technologies in patients' homes. *Journal of Biomedical Informatics*, 36:45–60.

Kintsch, Walter. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182.

Kittredge, Richard, Tanya Korelsky, and Owen Rambow. 1991. On the need for domain communication language. *Computational Intelligence*, 7(4):305–314.

Kittredge, Richard and John Lehrberger. 1983. *Sublanguages: Studies of Language in restricted semantic domains*. Walter DeGruyter.

Knight, Kevin and Daniel Marcu. 2000. Statistics-based summarization – step 1: Sentence compression. In *Proceedings of the American Association of Artificial Intelligence conference (AAAI'00)*.

Kucera, H. and W. Francis. 1967. *omputational Analysis of Present-Day American English*. Brown University Press.

Kukich, Karen. 1983. The design of a knowledge-based text generator. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'83)*, pages 145–150.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 68–73.

Kushniruk, Andre, Min-Yen Kan, Kathleen McKeown, Judith Klavans, Desmond Jordan, Mark LaFlamme, and Vimla Patel. 2002. Usability evaluation of an experimental text summarization system and three search engines: Implications for the reengineering of health care interfaces. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'02)*.

Lapata, Mirella. 2003. Probabilistic text structuring: Experiments with sentence

ordering. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'03)*, pages 545–552.

Lennox, Scott, Liesl Osman, Ehud Reiter, Roma Robertson, James Friend, Ian MacCann, Diane Skatun, and Peter Donnan. 2001. Cost effectiveness of computer tailored and non-tailored smoking cessation letters in general practice: randomised controlled trial. *British Medical Journal*, 322(7299):1396–1400.

Lin, Chin-Yew and Eduard Hovy. 2002. Automated multi-document summarization in NeATS. In *Proceedings of the Human Language Technology Conference (HLT'02).*

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using N-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'03)*, pages 150–157.

Lok, Simon and Steven Feiner. 2002. The AIL automated interface layout system. In *Proceedings of the Internationall Conference on Intelligent User Interfaces (IUI'02).*

Luhn, Hans Peter. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

M. Banko, V. Mittal, M. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'00)*, pages 318–325.

Mani, Inderjeet and Eric Bloedorn. 1999. Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1-2):35–67.

Mani, Inderjeet, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'99)*, pages 558–565.

Mani, Inderjeet, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 1999. The TIPSTER SUMMAC text summarization evaluation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 77–85.

Mann, William C. and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a function theory of text organization. *Text*, 8(3):243–281.

Marcu, Daniel. 1997. From discourse structures to text summaries. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88.

Marcu, Daniel. 1999. The automatic construction of large-scale corpora for summarization research. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR'99)*.

Maybury, Mark. 1995. Generating summaries from event data. *Information Processing and Management*, 31(5):735–751.

McClelland, J. and D. Rumelhart. 1981. An interactive activation model of context effects in letter perception: Part 1. an account of basic findings. *Psychological Review*, 88:375–407.

McCray, Alexa and Olivier Bodenreider. 2002. A conceptual framework for the biomedical domain. In Rebecca Green, Carol Bean, and Sung Myaeng, editors, *The Semantics of Relationships: An interdisciplinary perspective*. Kluwer Academic Publishers, pages 181–198.

McKeown, Kathleen. 1979. Paraphrasing using given and new information in a question answer system. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'79)*, pages 67–72.

McKeown, Kathleen. 1985. *Text generation*. Cambridge University Press.

McKeown, Kathleen, Regina Barzilay, David Evans, Vasilieos Hatzivassiloglou, Min-Yen Kan, Barry Schiffman, and Simone Teufel. 2002. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of the Document Understanding Conference (DUC'02)*.

McKeown, Kathleen, Desmond Jordan, and Vasileios Hatzivassiloglou. 1998. Generating patient-specific summaries of online literature. In *Proceedings of the Intelligent Text Summarization, AAAI Spring Symposium*.

McKeown, Kathleen, Karen Kukich, and James Shaw. 1994. Practical issues in automatic documentation generation. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'94)*, pages 7–14.

McKeown, Kathleen, Rebecca Passonneau, David Elson, Ani Nenkova, and Julia Hirschberg. 2005. Do summaries help? a task-based evaluation of multi-document summarization. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR'05)*.

McKeown, Kathy, Shih-Fu Chang, Jim Cimino, Steven Feiner, Carol Friedman, Luis Gravano, Vassileios Hatzivassiloglou, Stephen Johnson, Desmond Jordan, Judith Klavans, Andre Kushniruk, Vimla Patel, and Simone Teufel. 2001. PERSIVAL, a system for personalized search and summarization over multimedia healtcare information. In *Proceedings of the Joint Conference on Digital Libraries (JCDL'01)*.

Melamed, Dan. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130.

Mendonca, Eneida, James Cimino, Stephen Johnson, and Yoon-Ho Seol. 2001. Accessing heterogeneous sources of evidence to answer clinical questions. *Journal of Biomedical Informatics*, 34(2):85–98.

Meteer, Marie and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Proceedings of the International Conference on Computational Linguistics (COLING'88)*, pages 431–436.

Milosavljevic, Maria and Jon Oberlander. 1998. Dynamic hypertext catalogues: Helping users to help themselves. In *Proceedings of the Conference on Hypertext and Hypermedia*.

Moore, Johanna and Cecile Paris. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Journal of Computational Linguistics*, 19(4):651–694.

Morris, Andrew, George Kasper, and Dennis Adams. 1992. The effects and limita-

tions of automated text condensing on reading comprehension performance. *Information Systems Research*, 3(1):17–35.

Morton, J. 1969. Interaction of information in word recognition. *Psychological Review*, 76(2):165–178.

National Library of Medicine, 1995. *Unified Medical Language System (UMLS) Knowledge Sources*. Bethesda, Maryland. `http://www.nlm.nih.gov/research/umls/`.

Nenkova, Ani and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*, pages 145–152.

Okazaki, Naoaki, Yutaka Matsuo, and Mitsuru Ishizuka. 2004. Improving chronological sentence ordering by precedence relation. In *Proceedings of the International Conference on Computational Linguistics (COLING'04)*, pages 750–756.

Osman, Liesl, M. Adballa, J. Beattie, S. Ross, I. Russell, J. Friend, J. Legge, and J. Grahama Douglas. 1994. Reducing hospital admission through computer supported education for asthma patients. *British Medical Journal*, 308(6928):568–571.

Paice, Chris and Paul Jones. 1993. The identification of important conceptsin highly structured technical papers. In *Proceedings of the Conference on*

Research and Development in Information Retrieval (SIGIR'93), pages 69–78.

Pang, Bo, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'03)*.

Paris, Cecile. 1993. *User Modelling in Text Generation*. Frances Pinter.

Radev, Dragomir. 1999. *Language Reuse and Regeneration: Generating Natural Language Summaries from Multiple On-Line Sources*. Ph.D. thesis, Department of Computer Science, Columbia University.

Radev, Dragomir, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*.

Radev, Dragomir and Kathleen McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

Reape, Mike and Chris Mellish. 1999. Just what is aggregation anyway? In *Proceedings of the European Workshop on Natural Language Generation (ENLG'99)*.

Reiter, Ehud and Robert Dale. 2000a. *Building Natural-Language Generation Systems*. Cambridge University Press.

Reiter, Ehud and Robert Dale. 2000b. *Building Natural Language Generation Systems*. Cambridge University Press.

Riloff, Ellen. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*, pages 1044–1049.

Riloff, Ellen and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the American Association of Artificial Intelligence conference (AAAI'99)*, pages 474–479.

Robin, Jacques. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background: Corpus-Based Analysis, Design, Implementation and Evaluation*. Ph.D. thesis, Columbia University.

Saggion, Horacio, Kalina Bontcheva, and Hamish Cunningham. 2003. Robust generic and query-based summarisation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL'03)*.

Salton, Gerard, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–208.

Schank, Roger and Robert Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum.

Schiffman, Barry. 2005. *Learning to Identify New Information*. Ph.D. thesis, Columbia University.

Schulze-Kremer, Steffen, Barry Smith, and Anand Kumara. 2004. Revising the umls semantic network. In *Proceedings of the MEDINFO conference. Poster session*.

Shemtov, Hadar. 1993. Text alignment in a tool for translating revised documents. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL'93)*.

Siddharthan, Advaith, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the International Conference on Computational Linguistics (COLING'04)*, pages 896–902.

Siegel, Sidney and N. John Castellan. 1988. *Nonparameteric Statistics for the Behavioral Sciences*. McGraw-Hill.

Singer, Yoram and Robert Schapire. 1998. Improved boosting algorithms using confidence-rated predictions. In *Proceeding of Annual Conference on Computational Learning Theory*.

Soricut, Radu and Daniel Marcu. 2005. Towards developing generation algorithms for text-to-text applications. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL'05)*.

Spärck-Jones, Karen. 1999. Automatic summarizing: Factors and directions. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*. The MIT Press, pages 1–12.

Stock, Olivero and the ALFRESCO Project Team, 1993. *Intelligent Multimedia Interfaces*, chapter ALFRESCO: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration, pages 197–224. AAAI Press/The MIT Press.

Teufel, Simone, Vasileios Hatzivassiloglou, Kathleen McKeown, Kathy Dunn, Desmon Jordan, Sergey Sigelman, and Andre Kushniruk. 2001. Personalized medical article selection using patient record information. In *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA'01)*, pages 696–700.

Teufel, Simone and Marc Moens. 2002. Summarizing scientific articles: experimenting with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Tombros, Anastasios and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 2–10.

UM. 1994-2005. Proceedings of the conference on user modeling.

van Halteren, Hans and Simone Teufel. 2003. Examining the consensus between human summaries: initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL Workshop on Automatic Summarization (DUC'03)*, pages 57–64.

Wallis, Jerold and Edward Shortliffe. 1985. Customized explanations using causal

knowledge. In *ule-based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming*. Addison-Wesley Publishing Company, chapter 20, pages 371–388.

Williams, Sandra and Ehud Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proceedings of the European Workshop on Natural Language Generation (ENLG'05)*, pages 140–147.

Witbrock, M. and V. Mittal. 1999. Ultra-summarization: A statistical approach to generating highly condensed nonextractive summaries. In *Proceedings of the Conference on Research and Development in Information Retrieval Poster Session (SIGIR'99)*, pages 315–316.

Zukerman, Ingrid and Diane Litman. 2001. Natural language processing and user modeling: Synergies and limitations. *User Modeling and User-Adapted Interaction*, 11(1-2):129–158.

Appendix A

Scenario Example and Summaries under Different Conditions

This is an example of a scenario and its corresponding summaries generated under different conditions. The patient record was sanitized (dates and names are changed). The numbers in brackets in the summary texts link to specific sentences in the referenced input articles.

The Scenario

The Question

Could the atherosclerosis of the saphenous vein be prevented for this specific patient?

The Patient Discharge Summary

Note 1998-02-07
ADMITTED: 01/29/98
OPERATION DATE:

DISCHARGED: 02/07/98

HISTORY OF PRESENT ILLNESS: The patient is a 59 year old single man transferred from NY Hospital for evaluation and assessment of cardiac transplant.

Mr. XXXXX was a 59 year old man with coronary artery disease risk factors including hypertension, hypercholesterolemia, smoking, family history who first presented in 1988 with an anterior wall MI after reporting about 20 years of exertional angina.

In 1988 he underwent CABG with LIMA to his LAD, saphenous vein graft to PDA. Most recently two months prior to admission he noted increased exertional chest pain and increasing fatigue. One week prior to admission he noted increasing exertional angina, unresponsive to his conventional sublingual with rest angina. On 1/19 in the a.m. he had an episode of chest pain that lead to his current admission at NY Hospital. EKG on admission revealed a left bundle branch block. The patient was started on IV Heparin, Aspirin, IV Nitro. On 1/20 he underwent a repeat left heart cath which revealed a patent LIMA to his LAD, patent saphenous vein graft to the OM1 and OM2 and a 90% proximal lesion in the saphenous vein graft to the PDA with severe LV dysfunction and an EF of 15-20%. His right heart cath revealed a wedge pressure of 15 and cardiac output of 3.4.

He had intermittent chest pain and was evaluated for high risk PTCA and saphenous vein graft lesion. He underwent pericath placement of an intra-aortic balloon pump and PCA was planned. Repeat left heart cath on 1/28 revealed a new complete occlusion of the saphenous vein graft to the PDA and PTCA was felt to carry too great a risk and was deferred. At that time before intra- aortic balloon pump placement, his hemodynamics revealed an RA of 11, RA of 50/11 and PA of 56/35 and mean of 43. PA sat of 64% and wedge pressure of 35%. After intra-aortic balloon pump placement his PA pressures went down to 39/10 with a mean of 24 and a wedge pressure of 16. He was transferred on 1/29 for an elevation of LVAD or transplant to the CCU.

Only medical history was as above.

PHYSICAL EXAMINATION: On admission, a pleasant, pale appearing man, BP was 130/90, heart rate was 66. JVD, up/lying flat, lungs clear to percussion and auscultation. Heart exam was S1, S2 with a II/VI holosystolic murmur at the left upper sternal border. Abdomen: soft, no hepatosplenomegaly. Extremities: No edema. Neurologically intact.

DICTATED BY: B, M.D.
ATTENDING: M, M.D.

Generic Extractive Summary

The Summary

There was a small difference at baseline between the aggressive treatment group and the moderate treatment group (97.8% vs 98.9%, p = 0.02) that was not present at the later observation periods [1].

Among controls, menopause was associated with accelerated IMT progression (0.003 mm/year for premenopausal women vs. 0.008 mm/year for perimenopausal /b postmenopausal women for AVG IMT; p = 0.049) [2].

There was a tendency toward less atherosclerosis on baseline angiography in the aggressively treated group, but the angiographic end points evaluating change from baseline take account of baseline status [3]. Warfarin had no beneficial effect on the progression of atherosclerosis in the LMCA [3].

As evidenced by similar odds ratio estimates of progression (lumen diameter decrease >= 0.6 mm) and lack of interactions with treatment, a similar beneficial effect of aggressive lowering was observed in elderly and young patients, in women and men, in patients with and without smoking, hypertension, or diabetes mellitus, and those with and without borderline high-risk triglyceride serum levels [4]. This very low intensity anticoagulation showed no benefit with regard to angiographic and clinical outcomes [4]. There was no significant interaction between the lipid lowering and anticoagulation interventions [4]. The treatment effect of aggressive compared with moderate LDL-C lowering did not differ significantly among categories of all subgroups, as evidenced by the absence of significant interaction (P > 0.01) [4]. There were no interactions between treatment effect and all the subgroup categories [4].

There was no significant difference in angiographic outcome between the warfarin and placebo groups [5]. No significant differences in angiographic outcomes were observed between the warfarin and placebo groups [5]. The mean percentage per patient of grafts with occlusion or new lesions was significantly lower for patients assigned to aggressive treatment than for patients assigned to moderate treatment (Table 3) [5]. There were no significant differences among the study groups in the incidence of cancer or in deaths from cancer [5].

In the order of their importance they were: maximum stenosis of the graft at baseline angiography; years post-SVG placement; the moderate low-density lipoprotein cholesterol (LDL-C) lowering strategy; prior myocardial infarction; high triglyceride level; small

minimum graft diameter; low high-density lipoprotein cholesterol (HDL-C); high LDL-C; high mean arterial pressure; low ejection fraction; male gender; and current smoking [6]. Thus, the independent prognostic factors that make graft worsening more likely (in order of decreasing reduction in deviance) are: greater maximum stenosis of a graft; more years post-CABG; assignment to moderate (rather than aggressive) LDL-C lowering; prior myocardial infarction (MI); high triglycerides; smaller minimum diameter of the graft; low HDL-C; high LDL-C; high mean arterial pressure; low ejection fraction; being male rather than female; and being a current smoker [6].

Univariate predictors were restenotic lesion (odds ratio (OR): 2.47, confidence interval (CI): 1.13 to 3.85, P = 0.0003), unstable angina (OR: 1.99, CI: 1.27 to 2.91, P = 0.04) and congestive heart failure (CHF) (OR: 1.97, CI: 1.14 to 3.24, P = 0.02) for in-hospital MACE, and peripheral vascular disease (PVD) (OR: 2.18, CI: 1.34 to 3.44, P = 0.002), intra-aortic balloon pump placement (OR: 2.08, CI: 1.13 to 3.85, P = 0.02) and previous MI (OR: 1.97, CI: 1.14 to 3.25, P = 0.007) for late MACE [7]. Independent multivariate predictors for late MACE were restenotic lesion (relative risk (RR) 1.33, P = 0.02), PVD (RR: 1.31, P = 0.01), CHF (RR: 1.42, P = 0.01) and multiple stents (RR: 1.47, P = 0.004) [7]. Univariate predictors of in-hospital MACE included restenotic lesion, unstable angina and CHF (Fig. 1) [7]. Univariate predictors included PVD, IABP placement and previous MI; multivariate predictors of late MACE included restenotic lesion, PVD, CHF and placement of multiple stents (Fig. 1, Table 5) [7]. There was no difference in event-free survival for patients treated with a stent compared with those without a stent (Figure 3 and Figure 4) [7].

The Articles Indexed by the Summary

1. Quality of Life After Coronary Artery Bypass Graft . Results From the POST CABG Trial. Chest. Volume 126. 2004.

2. A dietary and exercise intervention slows menopause-associated progression of subclinical atherosclerosis as measured by intima-media thickness of the carotid arteries. American College of Cardiology. Volume 44. 2004.

3. Effect of an Aggressive Lipid-Lowering Strategy on Progression of Atherosclerosis in the Left Main Coronary Artery From Patients in the Post Coronary Artery Bypass Graft Trial. Circulation. Volume 104. 2001.

4. Aggressive Cholesterol Lowering Delays Saphenous Vein Graft Atherosclerosis in Women , the Elderly , and Patients With Associated Risk Factors . NHLBI Post Coronary Artery Bypass Graft Clinical Trial. Circulation. Volume 99. 1999.

5. The Effect of Aggressive Lowering of Low-Density Lipoprotein Cholesterol Levels and Low-Dose Anticoagulation on Obstructive Changes in Saphenous-Vein Coronary-Artery Bypass Grafts. New England Journal of Medicine. Volume 336. 1997.

6. Prognostic factors for atherosclerosis progression in saphenous vein grafts. American College of Cardiology. Volume 36. 2000.

7. Long-term clinical outcome and predictors of major adverse cardiac events after percutaneous interventions on saphenous vein grafts. American College of Cardiology. Volume 38. 2001.

Personalized Regenerated Summary

The Summary

Agressive lipid-lowering strategy and moderate low-density LDL-C lowering strategy were associated with atherosclerosis progression [1,2].

Predictors for atherosclerosis progression and graft worsening were stenosis of the graft, prior myocardial infarction, years post CABG, high triglyceride level, small minimum graft diameter, low HDL-C, high LDL-C, high mean arterial pressure, low ejection fraction, male gender, and current smoking [1,1].

There was no association between warfarin and progression of atherosclerosis [2,4].

Predictors of late MACE were unstable angina and CHF [3,3].

The Articles Indexed by the Summary

1. Prognostic factors for atherosclerosis progression in saphenous vein grafts. American College of Cardiology. Volume 36. 2000.
2. Effect of an Aggressive Lipid-Lowering Strategy on Progression of Atherosclerosis in the Left Main Coronary Artery From Patients in the Post Coronary Artery Bypass Graft Trial. Circulation. Volume 104. 2001.

3. Long-term clinical outcome and predictors of major adverse cardiac events after percutaneous interventions on saphenous vein grafts. American College of Cardiology. Volume 38. 2001.

4. The Effect of Aggressive Lowering of Low-Density Lipoprotein Cholesterol Levels and Low-Dose Anticoagulation on Obstructive Changes in Saphenous-Vein Coronary-Artery Bypass Grafts. New England Journal of Medicine. Volume 336. 1997.

The Articles not Indexed by the Summary

5. Aggressive Cholesterol Lowering Delays Saphenous Vein Graft Atherosclerosis in Women , the Elderly , and Patients With Associated Risk Factors . NHLBI Post Coronary Artery Bypass Graft Clinical Trial. Circulation. Volume 99. 1999.

6. Quality of Life After Coronary Artery Bypass Graft . Results From the POST CABG Trial. Chest. Volume 126. 2004.

7. A dietary and exercise intervention slows menopause-associated progression of subclinical atherosclerosis as measured by intima-media thickness of the carotid arteries. American College of Cardiology. Volume 44. 2004.

Appendix B

Input Articles Preprocessing

Article Genre Classification

Genre classification takes an article and classifies it into one of the four following categories: clinical study, review, letter to the editor, and case report. The classifier was trained on 2,700 articles and tested on 1,000 articles. To get the labels automatically, we took advantage of the meta-data field for publication type present for some articles indexed in the medical search engine PubMed[1].

Features included presence of an abstract, words in the abstract, number of words in the article. The genre classification achieves a general accuracy of 96% on the testing set. The specific classification for the "clinical study" category achieved 92.2% precision and 97.7% recal.

[1] `http://www.ncbi.nlm.nih.gov/PubMed/`

Article Clinical Task Classification

The clinical task classification takes as input a clinical study and determines its main clinical task: prognosis, treatment, or diagnosis. The training set (500 clinical studies) and the testing set (200 clinical studies) were manually annotated. The guidelines and the interface for the annotation are available at `http://www.cs.columbia.edu/~noemie/tas/clinical_task/`.

Features included words present in the abstract and the ratio of the different semantic types of the medical terms in the abstract. The clinical task classification achieved a general accuracy of 84.1%.

Appendix C

UMLS Modifications

Following is the list of the hierarchy of UMLS semantic types. The four-letter code is our label for each semantic type. Terms whose semantic types have the same labels were grouped together, and terms whose semantic type has the NULL label were filtered out.

NULL|Physical Object
- ORGM|Organism
  - ORG|Plant
    - ORGM|Alga
  - ORGM|Fungus
  - ORGM|Virus
  - ORGM|Rickettsia or Chlamydia
  - ORGM|Bacterium
  - ORGM|Archaeon
  - ANML|Animal
    - ANML|Invertebrate
    - ANML|Vertebrate
      - ANML|Amphibian
      - ANML|Bird
      - ANML|Fish
      - ANML|Reptile

ANML|Mammal

HUMN|Human

ASTR|Anatomical Structure

ASTR|Embryonic Structure

ASTR|Anatomical Abnormality

ASTR|Congenital Abnormality

ASTR|Acquired Abnormality

ASTR|Fully Formed Anatomical Structure

PART|Body Part, Organ, or Organ Component

ASTR|Tissue

ASTR|Cell

ASTR|Cell Component

ASTR|Gene or Genome

MANF|Manufactured Object

MANF|Medical Device

MANF|Research Device

DRUG|Clinical Drug

CHMC|Substance

CHMC|Chemical

CHMC|Chemical Viewed Functionally

CHMC|Pharmacologic Substance

CHMC|Antibiotic

CHMC|Biomedical or Dental Material

CHMC|Biologically Active Substance

CHMC|Neuroreactive Substance or Biogenic Amine

CHMC|Hormone

CHMC|Enzyme

CHMC|Vitamin

CHMC|Immunologic Factor

CHMC|Receptor

CHMC|Indicator, Reagent, or Diagnostic Aid

CHMC|Hazardous or Poisonous Substance

CHMC|Chemical Viewed Structurally

CHMC|Organic Chemical

CHMC|Nucleic Acid, Nucleoside, or Nucleotide

CHMC|Organophosphorus Compound

CHMC|Amino Acid, Peptide, or Protein

CHMC|Carbohydrate
CHMC|Lipid
CHMC|Steroid
CHMC|Eicosanoid
CHMC|Inorganic Chemical
CHMC|Element, Ion, or Isotope
CHMC|Body Substance
CHMC|Food
NULL|Conceptual Entity
NULL|Idea or Concept
TEMP|Temporal Concept
NULL|Qualitative Concept
NULL|Quantitative Concept
FCON|Functional Concept
PART|Body System
NULL|Spatial Concept
NULL|Body Space or Junction
PART|Body Location or Region
CHMC|Molecular Sequence
CHMC|Nucleotide Sequence
CHMC|Amino Acid Sequence
CHMC|Carbohydrate Sequence
NULL|Geographic Area
FIND|Finding
FIND|Laboratory or Test Result
FIND|Sign or Symptom
NULL|Organism Attribute
NULL|Clinical Attribute
NULL|Intellectual Product
NULL|Classification
NULL|Regulation or Law
NULL|Language
NULL|Occupation or Discipline
NULL|Biomedical Occupation or Discipline
NULL|Organization
NULL|Health Care Related Organization
NULL|Professional Society

- NULL|Self-help or Relief Organization
- NULL|Group Attribute
- NULl|Group
  - NULL|Professional or Occupational Group
  - NULL|Population Group
  - NULL|Family Group
  - NULL|Age Group
  - NULL|Patient or Disabled Group
- NULL|Event
  - ACTV|Activity
    - BEHV|Behavior
      - BEHV|Social Behavior
      - BEHV|Individual Behavior
    - BEHV|Daily or Recreational Activity
    - ACTV|Occupational Activity
      - ACTV|Health Care Activity
        - LPRO|Laboratory Procedure
        - DIGN|Diagnostic Procedure
        - THRP|Therapeutic or Preventive Procedure
      - ACTV|Research Activity
        - ACTV|Molecular Biology Research Technique
      - ACTV|Governmental or Regulatory Activity
      - ACTV|Educational Activity
    - ACTV|Machine Activity
  - PHNM|Phenomenon or Process
    - PHNM|Human-caused Phenomenon or Process
      - PHNM|Environmental Effect of Humans
    - PHNM|Natural Phenomenon or Process
      - PHNM|Biologic Function
        - PFUN|Physiologic Function
          - PFUN|Organism Function
            - PFUN|Mental Process
          - PFUN|Organ or Tissue Function
          - PFUN|Cell Function
          - PFUN|Molecular Function
            - PFUN|Genetic Function
        - DISS|Pathologic Function

DISS|Disease or Syndrome

DISS|Mental or Behavioral Dysfunction

DISS|Neoplastic Process

DISS|Cell or Molecular Dysfunction

DISS|Experimental Model of Disease

PHNM|Injury or Poisoning

Appendix D

Instructions for the Evaluation with Physicians

These are the instructions given to the subjects for the task-based evaluation with physicians. The interface used by the subjects, as well as the full text of the input articles are available at `http://www.cs.columbia.edu/~noemie/tas_eval`.

Thank you for participating. This study is part of a large scale project designed for clinicians which aims to provide easy access to knowledge at the point of patient care. We are investigating ways to present and index relevant information to the users using a computer system. The study is IRB-approved (#AAAA7549). It is fully anonymous.

We are going to present you with three scenarios. Each scenario should take approximately 20 minutes and is fully independent from the others. Each scenario consists of the following:

- A short admission note for a patient and the discharge summary of the patient's last visit to the hospital. Please take a minute to read both texts. You will also be provided with a simulation of a clinical question for this patient.
- A set of clinical studies that have been chosen to be relevant to the given question and the patient under care. Depending on the scenario you are presented with, the studies will be displayed on the screen as

  - a **List interface** - the title of the studies and publishing information are listed, or
  - a **Summary interface** - a short text summarizing the findings in the studies is presented. The text contains links to access the studies.

- A task to complete in 15 minutes. You will be asked to select in the studies all the **findings that are relevant to the given question and the patient under care**. A demo and more detailed instructions will be given to you shortly.
- A quick survey to fill out to get your impression of the task you completed.

▷ *If you agree to participate, please sign the informed consent form.*

Demo

▷ *Look at a demo of the system you will be interacting with for this study.*

▷ *If you have any questions about how to use the system, please ask them now.*

More on the task you have to complete

- We define a finding as **a result reported a clinical study**. Here are examples of what we consider findings:

  > The multivariable analysis identified previous diabetes (risk ratio 2.9), estrogen therapy (risk ratio 0.38) and left ventricular ejection fraction $< 40\%$ (risk ratio 3.9) as independent correlates of myocardial infarction or cardiac death after PTCA (Table 4).
  >
  > No associations were noted between noninvasive hemodynamic data and performance of invasive hemodynamic measurements.
  >
  > The most important predictor of developing atrial fibrillation was age (adjusted OR 3.20, 95% CI 2.99 to 3.43) (Table 5, Fig. 3). Other significant predictors (in decreasing order) included peak CK level, Killip class, heart rate, systolic blood pressure and height.

  Here are examples of what we do not consider findings:

  > The results are shown in Table 4.

The women in the population were significantly older than men.

Three patients died after three months.

- Select only the **findings** that you determine to be of **relevance to the given patient and the given question**. If you are presented with the List interface, you can select the findings by reading each study in turn. If you are presented with the Summary interface, you can use the same strategy, or you can read the summary first: if a sentence in the summary seems like a good candidate for selection, you can follow the reference in the summary which links to an actual sentence in a study. You can then select either the linked sentence in the study, or look around the sentence in the study for more relevant findings.
- Select any relevant finding that appears in any provided study. If two studies report the same findings, select **both findings** in each study. If the same study reports several times the same finding, select them all as well.
- While we are interested in your clinical expertise, we are only interested in your ability to identify **only the findings that are reported in the studies provided to you**, none from your own medical knowledge.
- Please **think aloud** during the task to explain what you are doing.
- You have 15 minutes to complete this task.

One last thing before starting

▷ *Please enter the following information.*

Gender: ☐ Male ☐ Female
Is English your first language? ☐ Yes ☐ No
What is your level of medical training (e.g., third year cardiology resident, fourth year medical student)?
How many years did you practice medicine?
What is your area of specialty?

▷ *You are now ready to start with the first scenario. If you feel confused by the above instructions, or if you have any questions, please let us know now.*

Appendix E

Technical/Lay Text Example

Here is the abstract of a clinical study published in the journal Lancet, as example of technical text. The second text is a summary of the same study published by ReutersHealth; it is targeted to a lay audience. The sentence numbers are given in brackets. A set of manually identified sentence pairs that convey similar information is then provided as an example of sentence alignment.

Technical Version from the Lancet

[1] The combination of fibrinolytic therapy and heparin for acute myocardial infarction fails to achieve reperfusion in 40-70% of patients, and early reocclusion occurs in a substantial number. [2] We did a randomised, open-label trial to compare the thrombin-specific anticoagulant, bivalirudin, with heparin in patients undergoing fibrinolysis with streptokinase for acute myocardial infarction.

[3] 17073 patients with acute ST-elevation myocardial infarction were randomly assigned an intravenous bolus and 48-h infusion of either bivalirudin (n=8516) or heparin

(n=8557), together with a standard 1.5 million unit dose of streptokinase given directly after the antithrombotic bolus. [4] The primary endpoint was 30-day mortality. [5] Secondary endpoints included reinfarction within 96 h and bleeding. [6] Strokes and reinfarctions were adjudicated by independent committees who were unaware of treatment allocation. [7] Analysis was by intention to treat.

[8] By 30 days, 919 patients (10.8%) in the bivalirudin group and 931 (10.9%) in the heparin group had died (odds ratio 0.99 [95% CI 0.90-1.09], P=0.85). [9] The mortality rates adjusted for baseline risk factors were 10.5% for bivalirudin and 10.9% for heparin (0.96 [0.86-1.07], P=0.46). [10] There were significantly fewer reinfarctions within 96 h in the bivalirudin group than in the heparin group (0.70 [0.56-0.87], P=0.001). [11] Severe bleeding occurred in 58 patients (0.7%) in the bivalirudin group versus 40 patients (0.5%) in the heparin group (p=0.07), and intracerebral bleeding occurred in 47 (0.6%) versus 32 (0.4%), respectively (p=0.09). [12] The rates of moderate and mild bleeding were significantly higher in the bivalirudin group than the heparin group (1.32 [1.00-1.74], P=0.05; and 1.47 [1.34-1.62], $p < 0.0001$; respectively). [13] Transfusions were given to 118 patients (1.4%) in the bivalirudin group versus 95 patients (1.1%) in the heparin group (1.25 [0.95-1.64], P=0.11).

[14] Bivalirudin did not reduce mortality compared with unfractionated heparin, but did reduce the rate of adjudicated reinfarction within 96 h by 30%. [15] Small absolute increases were seen in mild and moderate bleeding in patients given bivalirudin. [16] Bivalirudin is a new anticoagulant treatment option in patients with acute myocardial infarction treated with streptokinase.

Lay Version from ReutersHealth

[1] Heart attack patients treated with the blood-thinning drug bivalirudin were 30% less likely to have a second heart attack than those treated with the more traditional anticoagulant heparin, according to the results of a new study.

[2] Lead investigator Dr. Harvey D. White of Green Lane Hospital in Auckland, New Zealand, and colleagues report that bivalirudin (Angiomax) should be considered as a new treatment option for heart attack patients treated with streptokinase, another drug used to dissolve blood clots.

[3] The large international study, funded by bivalirudin's manufacturer, The Medicines Company, enlisted more than 17,000 patients in 539 centers across 46 countries. [4] Patients received either bivalirudin or heparin. [5] Both sets of patients also received aspirin and streptokinase.

[6] Bivalirudin was 30% more effective at reducing recurrent heart attack than heparin, translating to eight fewer heart attacks within 30 days for every 1,000 treated patients, the researchers report in the December 1st issue of The Lancet.

[7] The rate of death at 30 days after the initial heart attack was the same for both groups of patients and "small absolute increases were seen in mild and moderate bleeding in patients given bivalirudin," White and colleagues note.

[8] "Although there was no difference in the mortality rate, the 30% early reduction (within 96 hours) of recurrent heart attacks is impressive," Dr. Sidney Smith, chief science officer of the American Heart Association and professor of medicine at the University of North Carolina, Chapel Hill, said in a interview with Reuters Health.

[9] "This early reduction of recurrent heart attacks without an increased risk of major bleeding may contribute to the use of (additional treatments) in the further management of these patients," Smith added.

Manually Aligned Sentence Pairs

· Heart attack patients treated with the blood-thinning drug bivalirudin were 30% less likely to have a second heart attack than those treated with the more traditional anticoagulant heparin, according to the results of a new study.
· Bivalirudin did not reduce mortality compared with unfractionated heparin, but did reduce the rate of adjudicated reinfarction within 96 h by 30%.

· Lead investigator Dr. Harvey D. White of Green Lane Hospital in Auckland, New Zealand, and colleagues report that bivalirudin (Angiomax) should be considered as a new treatment option for heart attack patients treated with streptokinase, another drug used to dissolve blood clots.
· Bivalirudin is a new anticoagulant treatment option in patients with acute myocardial infarction treated with streptokinase.

· The large international study, funded by bivalirudin's manufacturer, The Medicines Company, enlisted more than 17,000 patients in 539 centers across 46 countries.
· 17073 patients with acute ST-elevation myocardial infarction were randomly assigned an intravenous bolus and 48-h infusion of either bivalirudin (n=8516) or heparin (n=8557) , together with a standard 1.5 million unit dose of streptokinase given directly after the antithrombotic bolus.

· Patients received either bivalirudin or heparin.
· 17073 patients with acute ST-elevation myocardial infarction were randomly assigned an intravenous bolus and 48-h infusion of either bivalirudin (n=8516) or heparin (n=8557), together with a standard 1.5 million unit dose of streptokinase given directly after the antithrombotic bolus.

· Both sets of patients also received aspirin and streptokinase.
· 17073 patients with acute ST-elevation myocardial infarction were randomly assigned an intravenous bolus and 48-h infusion of either bivalirudin (n=8516) or heparin (n=8557), together with a standard 1.5 million unit dose of streptokinase given directly after the antithrombotic bolus.

· Bivalirudin was 30% more effective at reducing recurrent heart attack than heparin, translating to eight fewer heart attacks within 30 days for every 1,000 treated patients , the researchers report in the December 1st issue of The Lancet.
· Bivalirudin did not reduce mortality compared with unfractionated heparin, but did reduce the rate of adjudicated reinfarction within 96 h by 30%.

· The rate of death at 30 days after the initial heart attack was the same for both groups of patients and "small absolute increases were seen in mild and moderate bleeding in patients given bivalirudin," White and colleagues note.
· Small absolute increases were seen in mild and moderate bleeding in patients given bivalirudin.

Appendix F

Instructions for the Sentence Alignment Annotation

Following are the instructinos given to annotators for the sentence alignment task. The annotators were presented with text pairs on paper and annotated sentences with a pencil. We show here the annotation instructions given for the Britannica texts; instructions were the same for the Reuters corpus, but with different examples.

Thank you for helping us.

The two texts you are presented with are from the Encyclopedia Britannica. They both describe the same city, but one is targeted for children (Text C) and the other for adults (Text A). As you will see, they contain a lot of information overlap. We ask you to match the sentences (one from each text) that convey the same information.

You should consider two sentences to convey the same information if they share at least one "clause". For instance, all of the following sentence pairs matched (the common clause is underlined here so that you understand why the sentences should be matched; in your annotation you simply have to mark sentences that match, not to identify the matching clause(s)):

"People have lived in the Lima area for thousands of years."
"The area around Lima has been inhabited for thousands of years."

"The Spanish conquistador Francisco Pizarro founded the city of Lima in 1535."
"The Spanish city of Lima was founded by Pizarro on January 6, 1535, which, being Epiphany, prompted the name Ciudad de los Reyes ('City of Kings')."

"It soon became Spain's main port in the region, and it was made the capital of the Viceroyalty of Peru."
"Although the name never stuck, Lima soon became the capital of the new viceroyalty of Peru, chosen over the old Inca capital of Cuzco to the southeast because the coastal location facilitated communication with Spain."

The following sentence pairs do not share any clause (even though they might share words) and, therefore, should not be matched:

"The city is also the country's leading port and center of commerce."
"And yet, as with so much of the city, the port facilities are old and inefficient."

"Adding to the complexity of daily living is a high rate of inflation, which hampers efforts to improve conditions."
"Many industries are also located in Paris."

One sentence can match several sentences, for instance the first sentence can be matched with the two following sentences:

"After World War II the population grew with amazing speed, mainly because thousands of families moved there from rural areas."
"Damascus' population of more than 1,200,000 represents a fivefold increase since 1945."

"After World War II the population grew with amazing speed, mainly because thousands of families moved there from rural areas."
"It has grown at a rate higher than that of the country as a whole due mainly to migration from rural areas."

You have unlimited time to complete this annotation.

Please let us know if you have any questions.