Short-term Event Tracking in Dynamic Online News

by

Jahna Clare Otterbacher

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Information)
in The University of Michigan
2006

Doctoral Committee:

Associate Professor Dragomir R. Radev, Chair
Professor Richmond H. Thomason
Associate Professor Steven P. Abney
Assistant Professor Soo Young Rieh
Professor Elizabeth Liddy, Syracuse University

Dedicated to my family and especially to my husband, Loucas.

ACKNOWLEDGEMENTS

My first acknowledgment must go to my advisor, Drago Radev, for supporting me during the course of my doctoral studies. I think that the thing I most enjoy and appreciate about working with Drago is his enthusiasm for research and teaching. I also want to thank Drago for believing in my work and my abilities, and for his encouragement. Secondly, I would like to thank the members of my dissertation committee, Steve Abney, Liz Liddy, Soo Young Rieh, and Rich Thomason, for being willing to spend their time on me and for sharing with me their thoughts on my research. Their input definitely helped me to strengthen and improve many aspects of this work, as well as my academic writing in general.

Many, many friends and colleagues at the School of Information helped me during my studies as well as made it a fun experience! In particular, I would like to thank the doctoral program manager, Sue Schuon, for being so helpful, organized and kind. Professors Judy Olson and Jeff Mason, who both served as the doctoral program chair during the four years that I was in the program, were also extremely encouraging and supportive during my studies. Finally, thanks to all of my fellow students at SI for their friendship over the years.

Last but not least, my family and friends have been very patient and understanding during the course of my studies. My parents have always been extremely supportive of me pursuing my educational goals. My sister, Marla Otterbacher, continues to inspire me with her own love of teaching and learning. Finally, my husband,

Loucas Louca, has been quite forgiving of me during the more stressful times, and has always encouraged my academic endeavors. Thank you so much to all of you!

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER I

Introduction

The World Wide Web ("the Web") has become one of the largest information and knowledge repositories in the world. As such, users rely on it as a convenient means to learn about topics of interest to them from a variety of sources and perspectives. However, finding relevant, quality information online is not a simple matter. In addition to the sheer size of the Web, another challenge for information seekers is that it is a dynamic environment. The Web is constantly in flux, with approximately 100 GB of textual material added each day [73]. A large portion of these updates comes from sources such as blogs, stock quotes and news stories that describe changes happening in the physical world. In this dissertation, I consider the problem of finding specific, factual information as it changes and is updated over time. In order to examine in depth how text on the Web conveys changes happening in the physical world, as well as how an information retrieval (IR) system might better support the user in searching for such information, I will focus on one particular genre of textual information on the Web - the breaking news story. In addition to investigating the properties of online breaking news stories that make it difficult for users to follow the information surrounding them, the goal of the thesis is to design and evaluate a system that is based on and can work with existing IR tools. However, in contrast to existing tools, the one developed in this thesis will have the specific goal of supporting the user in following the facts over time and across online news sources in a breaking story. I will refer to this task as "short-term event tracking" for dynamic news.

I have chosen to focus on online breaking news stories for several reasons. While news is convenient to collect and is freely available on the Web, it also exhibits many important features that challenge current IR systems operating in a dynamic information environment. In particular, a set of related stories collected online from different sources is dynamic information, because it is controlled by many independent news agencies. Therefore, one can often observe the same information portrayed in very different ways, due to the phenomenon of paraphrasing, the various biases of journalists and the fact that individual publications are written with a particular audience in mind. In addition, breaking news stories characteristically convey time-dependent information, describing events happening in the physical world that change - often rapidly - over time.

Figures 1.1 and 1.2 provide examples of the types of dynamic information that will be examined in this thesis. Figure 1.1 contains sentences extracted from articles that describe a major nightclub fire that took place in Rhode Island in February 2003. The sentences shown express information about the number of victims, and illustrate how information surrounding a story can change over time during the course of an investigation.

To contrast, Figure 1.2 shows sentences that were extracted from documents describing the April 2002 crash of a small plane into the tallest skyscraper in Milan, and concern the plane's origin and destination. This example shows that information reported at the same time about a particular fact can often change when news sources have access to different information, or when information sources have not yet reached a consensus as to what the ground truth is.

```
02/21/03 01:03 (ABC News)
A huge fire engulfed a Rhode Island nightclub during
a rock concert's pyrotechnics display, causing at least
10 deaths and 100 injuries, authorities said.

02/21/03 06:41 (CNN)
At least 26 people are dead after a concert's pyrotechnics
apparently ignited a massive fire that destroyed a
Providence-area nightclub late Thursday, officials said.

02/21/03 11:00 (MSNBC)
Fire engulfed a Rhode Island nightclub during a rock concert's
fireworks display, killing at least 60 people, authorities said Friday.

02/21/03 21:45 (CNN)
Ninety-six people died Thursday in a fast-moving fire at a Rhode Island
nightclub, Gov. Don Carcieri said Friday afternoon, adding that only a
handful of the bodies have been identified.
```

Figure 1.1: Dynamic information example: the known facts change over time.

```
04/18/02 13:17 (CNN)
The plane, en route from Locarno in Switzerland, to Rome, Italy, smashed
into the Pirelli building's 26th floor at 5:50 p.m. (1450 GMT) on Thursday.

04/18/02 13:42 (ABCNews)
The plane was destined for Italy's capital Rome, but there were conflicting
reports as to whether it had come from Locarno, Switzerland or Sofia,
Bulgaria.

04/18/02 13:42 (CNN)
The plane, en route from Locarno in Switzerland, to Rome, Italy, smashed
into the Pirelli building's 26th floor at 5:50 p.m. (1450 GMT) on Thursday.

04/18/02 13:42 (FoxNews)
The plane had taken off from Locarno, Switzerland, and was heading to Milan's
Linate airport, De Simone said.
```

Figure 1.2: Dynamic information example: sources report conflicting information.

1.1 Short-term Event Tracking: Finding Specific, Dynamic Information

There are many IR systems publicly available that aim to help keep users aware of the most current news on the Web. For example, services such as Google News [1]

[1] *http://news.google.com*

and NewsInEssence at the University of Michigan [2] [100] offer a tracking service, in which users receive an email when new articles about their subject of interest become available. In a sense, such services track information updates at the document level. However, users who seek a specific piece of information, such as a single fact or an answer to a question, need to read through the retrieved documents to find it. To contrast, online question answering systems such as NSIR[3] [98] accept a user's specific question of interest and then use Web documents to return a response, but do not track information change over time or between sources. In other words, systems such as NSIR implicitly assume that there is a single, best answer to the user's question, and do not allow the possibility for the correct answer to change with time.

There are many scenarios in which a user's information need requires a combination of the above technologies. When an event of great public interest happens, such as a terrorist attack or a natural disaster, Internet users are likely to turn to the Web to get answers to their questions, which might be related to their personal safety or that of a loved one. For instance, in the case of the September 11th terrorist attacks, many studies (e.g. [54, 92]) reported that Web news agencies were overwhelmed with demand during the attacks. In such emergency situations, users' questions of interest are likely to be specific, yet the answers to such questions are time-dependent (e.g. "Which areas have been affected?" "How many people were involved?"). Tools to support such information needs are needed, as the use of the Web for staying informed about world events is likely to continue [58], and the demand for customized information services, which allow the user to learn about a specific area of interest, is also expected to grow [9].

Another reason for developing a system for specific, dynamic IR is to support the

[2] $http://www.newsinessence.com$

[3] $http://tangra.si.umich.edu/clair/NSIR/html/nsir.cgi$

Information Synthesis problem [10], which is closely related to the short-term event tracking task. In contrast to the classical question answering setting in which the user presents a single question and the system returns a corresponding answer (e.g. as in the original TREC question answering setting [125]), here the user has a more complex information need. In the case of following changing information over time, such as in the emergency news story scenario, users might seek answers to a set of factual questions in order to understand the story better. In addition to conveying changing events over time, such stories are challenging in that they typically contain information about many sub-events. For example, in the Asian Tsunami story (December 2004), some important sub-events were the initial devastation of the tsunami, the relief effort, and the investigation into why there were few forewarnings of the disaster. Likewise, while some facts surrounding the story did not change (such as "Where did the tsunami first hit?"), other changed with time (e.g. "How many people have been confirmed dead?"). Therefore, in order to build IR systems that assist users in finding information that helps them fully understand a story or situation, such systems must be able to handle time or source-sensitive queries while at the same time permit the user to pose a wide range of questions.

Having motivated the development of an IR system that can support the seeking of factual, dynamic information in breaking news stories, the next sections will better position the work described in the current thesis. Specifically, Section 1.2 discusses the Web as a dynamic information environment, and will illustrate that breaking news is a dynamic information source. Section 1.5 illustrates why current IR applications, in particular text summarization and question answering systems, are not adequate in the context of the short-term event tracking problem. Finally, Section 1.6 states the specific goals of this work and outlines the five inter-related studies that will be presented in this thesis.

1.2 The Web as a Dynamic Information Environment

While it is difficult to estimate how large the Web is, what is clear is that its size is increasing at a geometric rate [18, 69]. It is also known that Web documents themselves are incredibly dynamic. In particular, young Web documents are typically unstable in that they are frequently modified, while older documents that have survived beyond a particular age tend to exhibit little change [60]. However, the likelihood of a page changing (in terms of its textual content) also depends on the type of site [85]. For example, pages on a university Web site may change less frequently over time than those on a news agency's site. This dynamic nature presents many challenges to Web-based IR applications, from search engines to question answering systems. For instance, for search engines using Web crawlers to index pages, one important question is how to estimate when a page has changed, so that the crawler can revisit and recache it [26, 11].

Teevan [121], in studying how people re-find information on the Web in pages that they have previously visited, uses the term *dynamic information* to refer to "any information that has changed in any way." To contrast, I will be concerned with dynamic information that is conveyed exclusively through text. Textual information in Web documents can be dynamic for a number of reasons. One is that content is controlled by many different agents rather than a by central authority. In the case of a user trying to follow the facts surrounding an emergency news story across time and from multiple news sources, this means that she is likely to see the same information expressed in a number of ways.

One more concern about seeking information in the dynamic environment of the

Web, which is related to the fact that its content is controlled by many different agents, is that of information reliability. While others have focused on the problem of detecting deliberate deception in online text (e.g. [133]), in the case of online news, bias and access to information are arguably more of a concern. I have already mentioned that, in the case of following online news, agencies have different biases that affect how they report events to readers. This means that they may often contradict other news sources about what the facts surrounding a particular event are. In fact, in Section 1.3, I will demonstrate in an initial corpus analysis of online breaking news, that this happens quite often, such that if users wish to learn the correct set of facts as soon as possible, one must follow several sources at once. Therefore, in addition to the ability to track how specific information changes with time, the ideal IR system to support short-term event tracking should also incorporate the notion that information may also vary by source.

1.2.1 Web documents

Since much research in information science has concerned the question of what the terms “information” and “document” mean, here I establish what is meant by the term “Web document” that I will use throughout the thesis. Since the onset of the digital age, information scientists have debated the issue of what exactly constitutes a document. While traditionally, a document noted a textual record, new digital technology has brought this concept into question, with some claiming that information documented in any medium or form should be considered a document [20]. In addition, since it is so easy to annotate or revise a document in digital form, documents have become much more fluid, so much that some information scientists have proposed the idea of the document as a performance at a point in time, rather than being a fixed object that remains the same across time. However, others have argued that digital documents are also fixed, noting that in order to edit a digital document, one must begin with some fixed version [71].

Brown and Duguid have noted that digital formats have promoted the "social life" of documents [113]. Unlike paper documents, Web (hypertext) documents allow for immediacy of inter-textual links and support interchanges between authors and readers. For example, Cronin and colleagues have studied how the Web has changed the nature of academic publishing and scholarly work in general [33]. In particular, they note that since the Web has "peculiar social properties," conversations that take place on the Web differ substantially from those that take place in standard academic (written) discourse. Web-based discussions are fluid and synchronous and can be archived easily and quickly.

In the current work, I view a Web document as only the *textual* content on a individual Web page, captured (or downloaded) at a given point in time. Particularly, a news "document" or "article" represents the respective author's account of a news story at a specific point in time. These documents have a social component, in as much as they could be rewritten, copied (in part or whole), corrected or continued by the same or another author. This processes is carried out over time, on the page or at a different location on the Web. This idea will be discussed in more detail in Section 1.4.1, which considers how journalists are trained to write about breaking news stories.

1.3 Online News as Dynamic Information

In order to illustrate the dynamic properties of online breaking news stories, I conducted an initial analysis of three large clusters of breaking news stories as reported by several Web-based news agencies.[4] The stories followed were the Columbia space

[4] Note that a more extensive and thorough corpus analysis of breaking news stories will be presented in Chapter III.

Story	Sources	Articles	Time span
Columbia	USAToday, CNN, MSNBC, Fox, Ha'aretz, BBC	48	36 hours
RI fire	MSNBC, CNN, ABC, CBS, Fox, BBC, Ananova, Lycos	43	48 hours
Milan	MSNBC, CNN, ABC, Fox, USAToday, La Stampa	56	24 hours

Table 1.1: Corpus of breaking news articles.

shuttle disaster (February 2003), the Warwick, Rhode Island nightclub fire (February 2003) and the crash of a small plane into a skyscraper in Milan (April 2002). Table 1.1 shows the attributes of each story's cluster of news articles.

First, I read the most recently published article in each cluster, and generated a list of ten important factual questions, that are central to understanding what happened in the stories. I tracked the evolution of these facts across all documents in each cluster. In particular, I studied the relative order in which questions were answered and how long it took answers to stabilize (for all news sources to report the same information). In addition, I counted the number of times the answer to a question changed before stabilizing to the correct answer. This is shown in Table 1.2. It should be noted that 6 of the 30 questions never settled during the time period that the story made headlines. For example, in the RI fire story, two questions remained unresolved - who was to blame for the incident and whether or not the number of people inside the building at the time exceeded the legal capacity.

Among the 24 questions that did stabilize, the distribution of the time required to do so was rather skewed, with 8 questions taking longer than 24 hours, and 14 requiring less than 12 hours. For example, questions relating to the cause of an incident or the number of casualties are likely to stabilize over a longer period of time, while details external to the incident, such as the weather at the time of the event, are likely to settle relatively faster.

In addition to the time to stabilization, another observation from the analysis is that certain facts in an evolving story are more volatile than others. For example, in the RI fire story, the answer to the question "How many victims were there?" changed 32 times before the correct answer was reported. The answer went from "at least 10," to "10 confirmed, actual feared much higher" to "several" to "at least 39" to "at least 60" and changed numerous times before reaching the final reported answer of "96 were killed."

Order	**Columbia shuttle breakdown**			**West Warwick, RI fire**			**Milan plane crash**		
1	victims	1.5h	0	sprinklers	9.75h	0	building height	3h	1
2	last contact	1.75h	0	fire code violation	12h	0	pilot killed	3.5h	0
3	terrorist act	1.75h	0	building description	15.5h	0	plane type	3h	1
4	explosion	2h	4	injuries	24.75h	22	weather	4h	0
5	place	2h	2	cause	25h	9	# passengers	4h	1
6	location of debris	4h	6	fireworks permission	35.5h	14	plane's origin	8.5h	12
7	indications of trouble	14h	0	victims	35.5h	32	victims	24h	18
8	cause	57h	8	number in club	NA	NA	injuries	33h	13
9	parts found	59h	3	who was to blame	NA	NA	cause	NA	NA
10	injuries on ground	NA	NA	over legal occupancy	NA	NA	# in building	NA	NA

Table 1.2: Relative order, time to stabilize and number of incorrect or partially correct answers before stabilization

In short, tracking the facts in an online breaking news story across time and in multiple news sources is challenged by a number of factors:

- In the first hours after the event, a number of contradictory reports appear in the newswire.

- After initial hesitation and contradictions, the different sources finally settle on the same answers for most (but not necessarily all) questions.

- Different question types "settle" on the stable version of the facts at different times. For example, the initial analysis suggests that questions related to the cause of an incident typically take longer to settle as compared to questions about when or where the incident occurred. Likewise, some questions may never be answered.

- In breaking news stories, the correct answer to a given answer may change with time, as agents in the world learn more about an event.

1.4 Short-term Event Tracking from Breaking News Stories: a Model

Here, I put forward a model of following the events surrounding breaking news stories that will guide the work in the thesis. The model combines what is known about how journalists write breaking news, along with some assumptions about how they, as agents in the physical world, observe and express facts about newsworthy events in text. The model also assumes that information seekers learn of the events of the world through the news texts they read from multiple online sources. The model illustrates my assumptions of how information is conveyed through text and is found and interpreted by information seekers using IR systems.

1.4.1 How news is written

Traditionally, journalists are trained to use the "inverse pyramid structure" when writing news stories [81]. In this style, an article should begin with a broad overview of the situation or event, followed later by the finer details of the story. Therefore, the main challenge for the writer is to rank the information according to importance, so that it can then be summarized in the leading sentence. Uko [123] claims that the inverse pyramid structure was born as a result of the commercialization of the telegraph. Editors encouraged writers to get to the point of a story quickly, so that sub-editors receiving stories over the telegraph could quickly edit them by automatically removing the latter parts, if necessary.

In the case of breaking news stories, a system of rewrites and follow-ups is typically used, in order to inform readers of the most up-to-date facts about a situation. Major news organizations are likely to have reporters on the scene, who collect information and then call it into the news room, where staff writers then prepare the story for press. Alternatively, they may produce stories from news received from other sources or from the wire [30]. In order to keep stories new to the reader, journalists are encouraged to play up additional newsworthy facts [50]. Follow-ups should feature all the new developments in a story, while at the same time including the background information of the original article. Also, new leads should be linked to the previously reported news with the writer leaving much of the original story unchanged. However, if no new facts are available when a follow-up is scheduled to be written, the story may be simply reorganized in order to freshen it.

While the above mentioned practices describe how journalists traditionally follow and write news stories, the popularity of obtaining news online has greatly affected the news business and how journalists create news. For example, the Forrester Group has noted that news delivered to readers via the Web or email has become more and more customized to the reader's interest and has predicted that it will continue to do so in the future [58]. They have also claimed that all types of news, from local to world news, will soon become available on demand and through a variety of media outlets [9]. Such predictions have led some journalists to claim that traditional writing practices such as the inverse pyramid structure are a thing of the past and that journalists are increasingly encouraged to find creative means to please readers [123]. To contrast, some believe that the demand for up-to-date, online news has had the opposite effect on news reporting. Kirsner states that the "breaking news dilemma," has caused many news outlets to increasingly rely on wire services between scheduled issues of their publication [59]. This may be particularly true of smaller agencies, as developing resources for covering breaking news entails significant expenses.

Finally, another important point about news reporting is that stories are always told from a particular perspective. Journalism itself has often been described as a set of cultural practices, in which one must make judgments about news worthiness, interpret information and meaning and use various linguistic and narrative techniques in telling a story [39]. As such, news reporting is never free of bias.

1.4.2 A two-layer, noisy channel model of breaking news reports

As previously discussed, this thesis will be concerned with short-term event tracking in online breaking news, in which a user wants to follow a set of facts of interest over time, as reported by multiple news agencies. Given how breaking news is covered by journalists, we can view the fact tracking problem as a two-layered noisy channel model. The model is illustrated in Figure 1.3. In the figure, an example is given regarding a fact of interest about a major breaking news story that occurred in February of 2003, the Columbia space shuttle disaster. The factual question illustrated is “What caused the disaster?”.

In the model, happenings or events occur in the physical world at different points in time. News agents (e.g. reporters, writers) obtain information about such events directly, by having reporters on the scene, or indirectly, by receiving reports from other sources, such as newswire. In addition, it is not necessarily the case that the news agents obtain this information at the same time, or that they receive the same information. The news agents, given the information they have available, form perceptions about the event or situation. This is the first noisy channel that information passes through.

The second layer of the model depicts the process of telling the news story. This is also a noisy channel as writers have unique styles, and may write for various audiences. As readers of the news, we can only observe the texts published by news outlets, but do not witness the happenings in the world directly. Therefore, we only

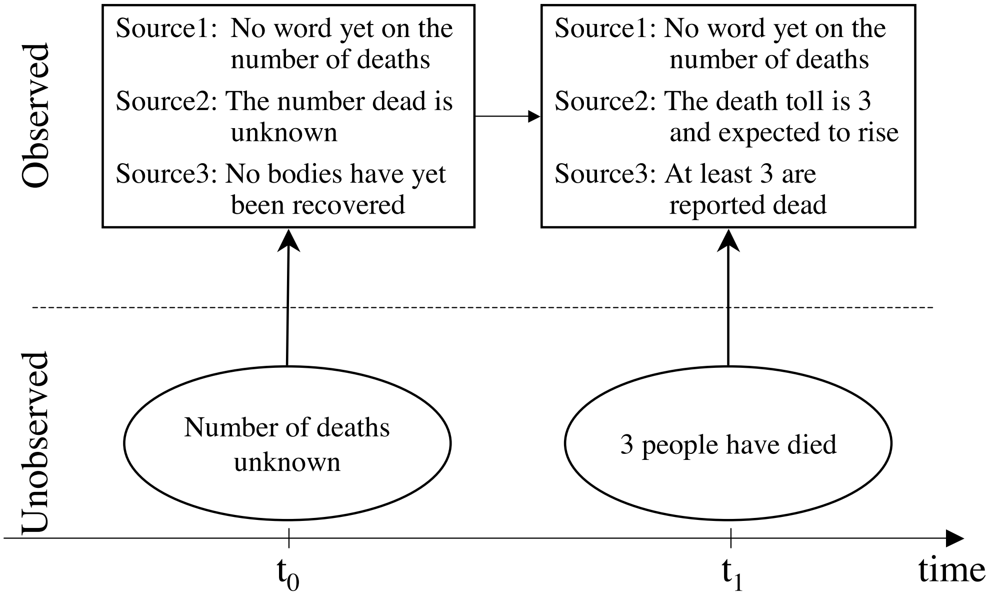

Observed
Unobserved
Source1: No word yet on the number of deaths
Source2: The number dead is unknown
Source3: No bodies have yet been recovered
Source1: No word yet on the number of deaths
Source2: The death toll is 3 and expected to rise
Source3: At least 3 are reported dead
Number of deaths unknown
3 people have died
t0
t1
time

Figure 1.3: A model of breaking news reports: "What caused the Columbia space shuttle disaster?"

know "the facts" surrounding a newsworthy situation as they are told by the news agencies. As depicted in Figure 1.3, news outlets often express the same information using different expressions (e.g. sources 1 and 2 at time t_0). In addition, the news agencies may attribute the same fact to different sources of information (e.g. sources 1, 2 and 3 at time t_1). In some cases, we can also expect to see contradictory information being reported across sources.

1.5 Information Retrieval Systems and Short-term Event Tracking

The previous sections have motivated the problem of following facts across time and in multiple sources in breaking news stories. In addition, the introduction argued that this task requires a combination of existing IR technologies. Specifically, it was argued that the ability to identify specific information, as in question answering systems, as well as the ability to identify new information over time, is necessary. Therefore, here I briefly discuss two areas in IR research that are closely related to the development of a system to support short-term event tracking, question answering and novelty detection, and note why such existing systems cannot support the short-term event tracking process. Finally, in Section 1.6, I will state the goals and the outline for the remainder of the dissertation.

1.5.1 Question answering and short-term event tracking

As previously mentioned in Section 1.1, question answering systems take as input a query in the form of a natural language question and return either a precise answer to the question or a set of documents that are likely to contain the answer. However, answering questions from dynamic information sources such as breaking news stories presents a number of challenges. First, they may express more than one answer to the question. To complicate matters, due to the presence of documents written by different authors, there may be more than one correct answer to a question or there may be some documents that contain incorrect answers. Finally, given that journalists are likely to use a system of updates and rewrites when covering breaking stories, another challenge is the presence of paraphrases, such that the same answer to a given question may be expressed in different ways.

Previously, in Table 1.2 in Section 1.3, I explained that it often takes time for facts in breaking news stories to settle down such that all news sources reach a consensus on what the ground truth is. Obviously, in the context of question answering, this means that what the "correct" answer is often depends on when the user asks it, and which news source's article is used to extract the answer. Figure 1.3 illustrates this for the question "Where was the plane's origin?" in the Milan plane crash story.

Time reported	**News source**	**Answer**
12:51 EST	CNN	Sofia, Bulgaria
13:17	CNN	Locarno, Switzerland
13:42	FoxNews	conflicting reports - Sofia or Locarno
13:42	MSNBC	Locarno, Switzerland
14:32	ABC	Lucerne Airport in Locarno, Switzerland
15:31	CNN	Magadino Airport in Rome
18:02	CNN	Magadino Airport near Locarno, Switzerland

Table 1.3: Answers to the question "From where did the plane originate?"

The answers to the question are shown with their respective publication time and news source.

Most question answering systems assume that there is one correct answer (or a "most correct" answer) to a given question that the user wishes to find[5]. The systems typically rank answers such that the top answer is deemed to be the most likely response to the user's question. To contrast, when answering a question from a set of articles describing a breaking news story, such as in the above example, I argue that there are two different approaches to answering the user's question. One approach is to build a system that can recognize when an answer has settled to its final answer, on which all sources agree. This approach would require the system to incorporate novelty detection, such that it could recognize if a new answer to the given question becomes available over time, or if an answer reported in a later document is the same as the previously reported answers. This will be discussed further in the next section. Finally, one major problem with this approach, is that as previously seen, some questions never reach a finalized answer.

A second approach to the problem is to return all identified answers to the input question back to the user. The answers should be reported with their respective publication times and news sources. In short, this approach, by returning all answers to the user, avoids having to incorporate novelty detection, but requires the user to

[5] Chapter II provides a review of current question answering technologies.

make judgments for him or herself as to what the finalized answer to the question is.

1.5.2 Novelty detection and short-term event tracking

Novelty detection is a relatively new IR task that has been described as the sentence-level analogue to the Topic Detection and Tracking (TDT) First Story Detection (FSD) problem [6]. While in FSD, documents are processed over time, and the documents describing a new (not previously discussed) news story are identified, novelty detection operates at the sentence level. Given a set of topically-related news articles, novelty detection systems process the documents in chronological order, first identifying the set of on-topic (relevant) sentences [116]. Next, the systems reprocess the list of relevant sentences, eliminating those containing redundant information, thus creating a list of novel sentences.

Novelty detection is clearly related to short-term event tracking in that it attempts to automatically recognize small text segments (sentences) that contain previously unseen information. However, recent research has questioned the feasibility of building accurate novelty detection systems at the sentence level. Some challenges that have been noted include the fact that there is typically low consensus between judges on identifying relevant and novel sentences [49, 116]. In addition, others have noted that novelty detection depends directly on the ability to first identify relevant sentences. However, detecting topically relevant sentences is a very difficult problem that is not yet being done to a high degree of accuracy [7]. Nonetheless, in the current thesis, I will consider the relationship of novelty detection to short-term event tracking. Chapter V considers in detail the feasibility of implementing fact-focused (rather than the previously attempted topic-focused) novelty detection at the sentence level.

1.5.3 Current approach: event tracking at the sentence level

As previously mentioned, in the current work, I will use existing IR tools to create a system that is specifically designed to support users in the short-term event tracking task. In particular, I will be using two state-of-the-art systems, the MEAD extractive summarization environment [101] and the NSIR question answering system [98]. As discussed in Section 1.5.2, finding novel information over time at the sentence level has not yet been done satisfactorily. However, one hypothesis is that topic-focused relevance and novelty judgments are very context-dependent and are difficult to define [7]. At the same time, as will be discussed in Chapter II, many IR systems operate at the sentence level and stand to benefit from the development sentence-level novelty detection methods. Therefore, the goal of the current work is to approach short-term event tracking at the sentence level of granularity, by introducing a fact-focused notion of relevance and novelty, that will be discussed in detail in Chapter V.

1.6 Thesis goals and outline

The current dissertation has two central goals:

1. To better understand the online, breaking news story as a source of dynamic information, and to characterize its challenges for information retrieval systems.

2. To develop and evaluate a system to support the short-term event tracking problem based on existing tools.

The next chapter, Chapter II, presents a survey of related research in the areas of information retrieval and natural language processing that is related to the problem of building the proposed IR system. In the survey of the literature, I will discuss existing systems that help users find specific pieces of information (e.g. question answering systems and question-focused summarization, which can be used to find particular facts) as well as work that has addressed the novelty detection problem at different levels of granularity (e.g. the document level versus the sentence level). I will make the case that a system to support short-term event tracking should be query-sensitive and should also account for the possibility that information changes over time and across different sources. I will also review text processing techniques that have been applied to the problem of distinguishing novel from similar information, and will highlight the fact that most methods are still lexical in nature (i.e. classify textual units as being similar if they share common words).

Following the survey of the literature, I will present five inter-related studies that address the two main goals. The first two studies, described in Chapters III and IV, explore and characterize the challenges of the breaking news story from an information retrieval perspective. The first study is an empirical analysis of a set of breaking news stories, which were collected from online news agencies. Each news story in the collection had a set of relevant documents from various news sources, as well as a set of factual questions that subjects deemed as being key to understanding the given story. The news articles about a story were manually annotated at the sentence level for the presence of answers to each of the questions. In order to see how answers to a given question evolve over time, I conducted a semantic analysis in which each answer to a question was compared to the finalized answer. In this case, the finalized answer is the "settled" answer, on which all (or the majority) of the news sources eventually agreed. I found that across the entire corpus, which consisted of 9 breaking news stories and 2,437 answers to the sets of questions, only 14.6% of the answers represented settled answers. In other words, 85% of the reported answers to factual questions were corrected or updated at a later point in time. Another challenge that was highlighted in this study is that many subtle but non-trivial relationships exist between the different answers that are reported to the same question over time and across sources. For instance, different news agencies may report contradictory answers at the same point in time. Another example is that the reported answers may differ as to very specific details (e.g. one source may report the exact time of an incident while another may give a more general statement of when it occurred such as "this afternoon.")

While Chapter III examined how factual answers to questions surrounding a breaking story evolved over time at the sentence level (i.e. each sentence in each article related to a particular story was marked for the presence or absence of an answer to a given question), the analysis in Chapter IV took place at the document level. In particular, the goal was to see whether or not there was evidence that sets of breaking news articles evolve over time. To answer this question, I fit a biologically inspired phylogenetic model to each cluster of news articles. In phylogenetic models, it is assumed that a set of species (in this case, a set of topically related news articles) evolves over time from a common ancestor, with mutations (e.g. changes in particular facts) occurring at various points in time. The phylogenetic model itself shows the most likely evolutionary history of the documents. I evaluated the fit of the model with respect to how well one can use it to infer chronological relations between the documents (as verified by their publication times). I found evidence of "evolution" in sets of documents that were published within short time periods of one another (e.g. within hours of one another). However, for cases where a news story was told over a longer period of time, the phylogenetic model did not fit as well. While the phylogenetic study did not directly provide practical implications for the building of my event-tracking IR system, it showed that breaking news stories unfold differently over time. In particular, it appears that for some stories, the assumption of evolving from a common starting point (e.g. set of initial facts) is valid. However, for other stories, there is not such a common “ancestor,” with news sources initially publishing a variety of facts before finally converging on the grounded set of facts.

In the next chapter, Chapter V, I tested the hypothesis that sentence level novelty detection might be more feasible in the fact-focused setting, as compared to previous research in which the goal was to detect sentences that are relevant to a more general, topic query and that contain previously unseen information [116]. As mentioned in Section 1.5.2, past studies have reported that novelty at the sentence level is too subjective and context-dependent. Therefore, I evaluated the interjudge agreement on the task of identifying sentences that are relevant (i.e. contain an answer for) a given factual question. I also evaluated the agreement on a second step, which was to determine which of the relevant sentences were also novel. In contrast to previous findings, which reported that judges did not agree on sentence-level relevance judgments [116, 7, 111], I found a high level of agreement between judges on finding sentences relevant to a factual question. However, there was a low level of agreement on finding which sentences provided novel answers to a question. In terms of the novelty problem, the experiments suggested that novelty is difficult to operationalize at the sentence level. This conclusion is consistent with the findings in Chapter III, that there are many subtle semantic relationships that hold between sentences containing answers to a question, which may make sentence level novelty judgments too subjective. In terms of practical implications for building a system, the results suggest that when working at the sentence level, it is more fruitful to concentrate on the automatic identification of sentences that are relevant to a question, rather than on identifying novel sentences automatically.

Given the findings from Chapter V, I propose a design for an IR system in Chapter VI that does not involve novelty detection. Rather, the system, when given an input set of documents related a breaking news story and a factual question of interest, displays all extracted answers to the question along with their respective publication times and news sources. The system is built using components from two state-of-the-art IR systems, the MEAD text summarizer [101] and the NSIR question answering system [98]. In particular, I focused on developing a question-focused sentence retrieval method using the MEAD framework [86]. As discussed in Chapters II and III, one challenge for building a system is the presence of paraphrasing, such that reported answers to questions can express the same meaning, but use very different words to do so. This means that, when retrieving sentences that are relevant to an input question, if one simply looks for sentences that are similar to the question, more lexically diverse sentences (i.e. paraphrases of the sentences that are similar to the question) will be missed. In order to address this problem, in Chapter VI, I use a technique that exploits both the similarity of the sentences to the input question, as well as the similarities between the sentences themselves. Once the set of relevant sentences is identified, NSIR is then used to extract answers from the sentences passed onto it from MEAD. In addition to presenting the overall system architecture, Chapter VI also describes the sentence retrieval method and experiments using the method on a corpus of breaking news stories in detail.

Finally, Chapter VII describes a task-focused user study designed to evaluate how well the system facilitates the finding of time and source-sensitive information from online breaking news stories. Subjects used three different systems (one baseline as well as two configurations of the new system) to complete three different timed information searching tasks. A discussion of the study's results will demonstrate that while there are no performance differences between the three systems, one configuration of the new system significant reduces the users' search efforts as compared to the other two systems. In particular, when the users complete the task using a system that returns the top 20 most relevant sentences to an input question, they must search through fewer source news articles in finding the answers to questions, as compared to the other systems.

CHAPTER II

Survey of Related Work

This chapter surveys previous information retrieval and natural language processing research that is related to the problem that was motivated in Chapter I - that of designing a system that supports the tracking of specific, dynamic information across time and from texts published by different sources. I will first discuss some existing IR systems. Section 2.1 surveys systems that are able to retrieve specific pieces of information, including question answering and question-focused text summarization systems as well as information extraction. Next, in Section 2.2, I survey some existing systems designed to track changing information over time. Section 2.3 will focus on previous approaches to detecting change or dissimilarity between textual units, and will emphasize how this problem has been approached at different levels of textual granularity. Following that, I will discuss previous research on the automatic detection of semantic relationships between textual units in Section 2.4, with a focus on discourse and temporal relationships. Finally, I will conclude the literature survey by discussing some of the key challenges in terms of building the proposed IR system, that are suggested by the review of the previous relevant work.

	Open-domain?	**Full Web access?**	**Source sensitive?**	**Time sensitive?**
Question answering				
Webclopedia	Y	Y	N	N
Ionaut	Y	N	N	N
SMART	N	N	N	N
MURAX	N	N	N	N
Falcon	Y	N	N	N
TextRoller	Y	N	N	N
NSIR	Y	Y	N	N
Mulder	Y	N/A	N	N
Focused summarization				
Definition summarization	Y	Y	N	N
Snippet retrieval	Y	Y	N	N
Information extraction				
MUC systems	N	N	N	N
SUMMONS	N	N	Y	Y
IE from document threads	N	N	Y	Y

Table 2.1: A Survey of IR systems for specific queries.

2.1 IR Applications for Finding Specific Information

In this section, I will discuss many of the IR systems that have been built to support the retrieval of specific information (e.g. facts) from textual documents. The three categories of systems to be discussed are question answering systems, focused text summarization and systems for information extraction. Table 2.1 summarizes the capabilities of several such systems with respect to whether or not they are open-domain, whether they use documents from anywhere on the Web to find answers, and whether or not they incorporate the notion that answers can be source-sensitive and time-sensitive into the answer-finding process.

2.1.1 Question answering

As previously noted, question answering systems address the fine-grained information needs of users, by returning an answer, or a document containing an answer, to a user's factual question of interest. In recent years many question answering systems have been developed, and here I classify systems into two broad categories: those developed to perform on a restricted corpus of texts and Web-based systems.

TREC Systems

Much of the recent developments in question answering research is due to the Text REtrieval Conference (TREC) question answering evaluation [125], in which participants use the TREC corpus (2 GB of text) to develop and test systems that find answers, or short passages containing the answers, to factual questions. Several new techniques and approaches to Q&A have been developed within the TREC framework. For example, in "predictive annotation" [94, 106], documents in the collection are first marked up with labels describing the question types for which they could potentially provide an answer. Next, passages that might contain the answers for an input question are retrieved, and the candidate answers are extracted from the passages. Finally, answers are ranked in terms of their likelihood of being correct, using various heuristics. To contrast, other systems such as Hovy and colleagues' Webclopedia [55] and the Ionaut system developed by Abney and colleagues [1] make heavier use of NLP techniques. Webclopedia parses input questions in order to create a query for the retrieval of relevant documents. The retrieved documents are then segmented and the small segments are ranked for relevance to the question. Potential answers are extracted from the passages and the questions are assigned a question type, according to a set of manually created rules. Finally, the extracted answers are reranked, according to how well they fit the question's type. On the other hand, the Ionaut system makes use of named entity techniques. First, given the input question, passages likely to contain the respective answer are retrieved. Next, the named entities in the passages are identified. The question and the entities are classified using a predefined set of categories, and the entities that are not of the type required by the question are eliminated. Finally, the remaining entities are ranked according to word frequency and proximity information.

Some of the TREC Q&A systems have used more semantic information for finding answers to input questions. While Clarke and colleagues [27] also used passage retrieval techniques in the initial step, they then reranked passages using semantic match information between the type of question posed and the terms contained in the candidate passages. In particular, they used a question parser based on WordNet [80] to assign the question to a semantic category. The Falcon system [48] also makes use of WordNet, in order to reformulate the input question, adding more semantic information. In retrieving relevant passages, named entity techniques are also used. Answers in the retrieved passages that match the question's respective type are extracted and are put through a test based on abductive reasoning. The answers passing the test are then kept.

One TREC system that differs in approach from all of the others previously discussed is the TextRoller system [117], which was the top-scoring system in the TREC 10 evaluation. Rather than using NLP techniques, TextRoller uses a new approach called the pattern-based approach. For each question type, various answer patterns are defined. They are then used for pattern matching from the texts to find candidate passages, as well as in selecting and ranking the answers.

Web-based systems

Rather than answer questions from the documents contained in a particular corpus, as in the TREC systems, Web-based systems attempt to answer input questions in the much larger context of the Web. One of the first systems developed for question answering from the Web was START [57]. However, while it answered questions from online information, it relied on a knowledge base in order answer queries in a restricted domain (geography and the MIT InfoLab). Similarly, MURAX [63] used an online encyclopedia to answer users' trivia questions.

One direction in research on Web-based Q&A, is to attempt to use currently existing search engines as a first step towards question answering, as suggested by Radev and colleagues [104]. To this end, several efforts, such as those put forward by [2] and [46] focused on formulating a search engine query given a question of interest, in order to obtain the optimal results in terms of question answering. While Agichtein and colleagues concentrated on techniques for learning the ideal query transformation process for use with specific search engines, Glover and colleagues tried to develop a means for adding more domain specific information to input questions in order to improve the hits from search engines.

The Mulder system [66] is an open-domain Web-based system that uses techniques similar to those used by TREC systems, such as syntactic parsing of the user's input question as well as the classification of questions as to their expected answer types (e.g. nominal, numerical, temporal). However, Mulder is no longer available on the Web. As noted on the Mulder Web page, approaches to Q&A that rely on deep NLP techniques such as parsing, are very slow and are not practical for use on the Web. To contrast, the NSIR system [99], another general-purpose Web-based system, substitutes such time-consuming modules with rule-based classifiers and a technique known as "probabilistic phrase reranking." NSIR also uses query modulation to retrieve relevant documents from three Web-based search engines, Google[1] , Northern Light[2] and All the Web[3]. This is followed by sentence retrieval, in which sentences containing answers are automatically identified, answer extraction and finally, answer ranking. According to [99], NSIR obtains "reasonable performance," while at the same time running fast enough to offer its services on the Web.

As is summarized in Table 2.1, the TREC systems as well as some of the Web-based systems are able to handle questions from a wide-variety of domains. However, as noted, I am not aware of any Q&A system that attempts to address the fact that many questions posed by users, such as those surrounding emergency events in the world, are sensitive to the time the source article was published as well as who wrote it. In short, while they offer fine-grained information retrieval, current Q&A systems return the answer (or set of answers) that they deem to be the most relevant to a user's input question, rather than necessarily providing the most "correct" answer currently available, as of the time the question is posed to the system.

2.1.2 Answer-focused summarization

Motivated by the idea that a user's information need is often best described as a question or a set of questions, answer-focused summarization was one of the tasks in the 2003 Document Understanding Conference (DUC) [88]. The goal is to produce a summary of one document, or a set of multiple documents, that contain an answer to the user's question. In this sense, it is similar in spirit to question answering, in that it attempts to address a specific information need, rather than giving an overview of all of the information contained in the input document(s), as in generic

[1] *http : //www.google.com*
[2] *http : //www.northernlight.com*
[3] *http : //www.alltheweb.com*

summarization.

Wu and colleagues [128] present a technique called “Snippet Retrieval,” in which a user’s question is sent to Web-based search engines such as Google to retrieve short passages describing the documents on the hit list. They then identify all of the query words that appear in the passages, extract windows of a variable size around the query words, and then order the extracted windows with respect to the number of query words they contain. Finally, they concatenate the windows of text until the desired summary length is reached. In their experiments, in which they compared this technique to the passages that Google provides of each document, a window size of 4-5 words was optimal in terms of answering the most questions in their data set. In addition, their technique outperformed the Google baseline. However, it should be noted that their technique finds answers on a document-by-document basis and does not incorporate the notion of time-dependency.

Another work of interest is that of Cui and colleagues [35], who focused on producing summaries to answer definitional questions such as “Who/what is X?” where X is a person, organization or term of interest. In their system, input sentences (related to a person or term) are ranked with respect to their centrality to the topic (using centroid words as in [101]). Next, given a small set of labeled definition instances for training, the system learns soft matching patterns for deciding whether or not sentences are definitional. Then, the sentences are reranked in order to incorporate the weight of the pattern matching along with the centroid weight. Finally, in selecting the sentences for the resulting summary, they use the concept of Maximal Marginal Relevance [23] in order to select high-ranking yet non-redundant sentences. The system performed well both on the standardized TREC corpus as well as on crawled online news articles. However, as noted, the system focuses specifically on answering definitional questions, rather than supporting more specific questions.

Finally, there has been some research on producing multi-document summaries that are focused on multiple questions of interest [83]. This system used a question answering engine to assign a score to each input sentence with respect to each input question, which are then combined to create an overall score for sentence ranking. However, the system is motivated by the need for users to express their query through a set of questions, rather than focusing on retrieving information about one specific question.

2.1.3 Information extraction

In contrast to question answering systems or focused-summarization systems, which can often handle questions that are input spontaneously by users, traditional information extraction (IE) systems are designed to find answers to a predefined set of questions [32]. This is also done on a document-by-document basis. Perhaps the best known research initiative in the area of IE was the Message Understanding Conferences (MUC), a series of tasks and evaluations sponsored by DARPA during the 1990s. The main goal was to support the development and evaluation of systems that could process news articles from specific domains, extracting salient, important pieces of information. To this end, domain specific templates, called scenario templates, were developed that identified the slots of information to be filled in by the IE systems.

Table 2.2, adapted from [32], shows an excerpt of a template used in the MUC-4 terrorist task, in which systems were to identify particular information about terrorist incidents from news texts. Slots in the template to be filled in by the systems include those that require strings from the text (such as the location of the terrorist incident) and set fills, in which one of a set of categorical answers is chosen. For instance, in

Template Slot ID	Fill Value
Incident: Date	07 Jan 90
Incident: Location	Chile: Molina (city)
Incident: Type	Robbery
Incident: Stage of execution	Accomplished
Incident: Instrument type	Gun

Table 2.2: Excerpt from MUC terrorism template

the example, the type of incident, robbery, was chosen from a set of categories such as assault or murder.

The template-based understanding systems, having been developed, evaluated and refined over a long period of time, are quite robust in their ability to process large quantities of text. In fact, the MUC Web site[4] reports very high rates of reliability for the state-of-the-art systems on various extraction tasks (e.g. 90% for named entities, 80% for specific attributes of entities, 70% on finding facts, and 60% for specific events). The template-based approach has also been used successfully in single-document summarization [22]. However, since these systems process documents independently, they cannot identify semantic relationships that hold between multiple texts, such as paraphrase or contradiction. As such, they do not incorporate the notion that some extracted answers may be more reliable or more informative than others. In addition, since they process single documents at a given point in time, they cannot detect changes in information over time.

IE and dynamic information

While the MUC systems themselves do not try to reflect changing information over time, a summarization system called SUMMONS [105] used the MUC templates as input for generating summaries of multiple documents. The input documents were related to the same topic, terrorist events. In order to highlight how knowledge of the

[4] $http://www.itl.nist.gov/iaui/894.02/related\_projects/muc$

facts and the perspectives of an event change over time, SUMMONS uses operators that can combine the information contained in a set of templates, extracted from multiple documents. However, due to its reliance on domain-dependent semantic templates, the system cannot process documents that are not related to terrorism.

Another area of research has considered the extension of IE techniques to apply to multi-document threads (i.e. topically related documents seen over time). Citing the fact that in many work-flow scenarios, a single conversation or transaction between multiple individuals takes place over several natural language texts, [79] considered the extraction of information from sets of emails. The corpus of emails studied concerned student applications to a particular graduate program, such that the values of interest included details such as applicant name, assigned identity number, degree type and matriculation date. An algorithm for finding such information in single texts was first trained on the corpus. Next, the learned rules were applied to N documents in a given thread of emails, creating a candidate set. Each extracted field also received a confidence value reflecting how likely it was to be the correct answer. In processing the texts in chronological order, values in the template were replaced with the value extracted from the current text, only if its respective confidence value exceeds that of the value currently in the template.

In this context, it is assumed that there is one correct answer to each field in the template, such that information does not change with time. Rather, the processing of multiple documents allows for additional chances to extract the correct values for each slot in the template. In this way, the system does incorporate the notion of information reliability. However, it does not take into consideration the case of dynamic information, such that the “correct” answer changes by time or according to which source provides the information.

2.2 Systems for Detecting Information Change Over Time

I now turn to discussing systems that are explicitly designed to detect changing information over time. In contrast to Section 2.3, which discusses more generally, techniques for the detection of textual dissimilarity at different levels of granularity, the systems discussed in the current section all address information that *changes over time*. Also, as compared to many of the systems discussed previously in Section 2.1, those described below do not allow the user to find specific information, but rather, aim to detect in general, when new information has become available, in order to alert the user.

2.2.1 Changes in Web pages

An early system proposed and developed by [102] called Rendezvous aims to keep users informed when Web-based information of interest to them changes or is updated. Rendezvous operates by accessing the user's hotlist, a list of URLs indicating a set of Web resources that they have previously used and are likely to want to reaccess in the near future. The system then checks the relevant servers regarding the creation and modification dates of the pages, in order to see if anything has changed. Finally, Rendezvous notifies the user via email of the relevant changes. While the user can specify the frequency of the reports to be received from the system, she cannot configure the system to check for specific types of information updates.

To contrast, the AIDE system [12, 36], combines Web page tracking and user notification with versioning and comparison of pages. It incorporates the *HtmlDiff* tool, that compares two HTML pages (e.g. the same page accessed at two different points in time), and can detect subtle differences between them. More specifically, rather than detecting differences in formatting or page layout, HtmlDiff focuses on comparing the content of pages. To compare a given Web page to the previous version, in order to determine what has changed, AIDE views the HTML documents as sequences of sentences, with any tags as sentence-breaking markups. It then attempts to align a sentence in the first document with one in the second document, and sentence-breaking markups are matched to one another as well. The longest common subsequence (LCS) metric is used to compare the two documents. In short, the problem is to find the common subsequence of the two documents that has the longest length. However, the common subsequence does not have to be contiguous. This is similar to the comparison algorithm used by the UNIX utility, *diff*, except that in HtmlDiff a token is a sentence or a sentence-breaking markup (tag), while *diff* operates at the word level. Any tokens that are not in the LCS represent changes that have occurred between the earlier and later versions of the HTML document.

AIDE can be used for applications such as collaborative editing (i.e. determining particular changes on a page from one version to the next) and for coordinating distributed work. However, like Rendezvous, AIDE does not accept queries from the user about specific information changes. Rather, the Web page is the unit of analysis. In addition, AIDE would not be able to discern the case where a Web page author has simply refreshed something on the page, by paraphrasing or rewording something from one version of the page to the next, from the case where the author has actually changed the meaning of the information conveyed. Therefore, in domains such as news, where paraphrasing occurs more often than not, AIDE would likely return more changes to the user than he or she would be interested in seeing.

2.2.2 Topic Detection and Tracking

The Topic Detection and Tracking (TDT) research initiative addresses issues related to the organizing of streams of broadcast news by event or topic. The TDT community has focused on several research questions, including how to automatically identify discrete news stories in a stream and how to detect the onset of a new story or topic, First Story Detection (FSD) [3]. Another TDT task that is related to FSD is Link Detection (e.g. [25]), in which the goal is to automatically determine if two input documents describe the same topic or story. The TDT systems are intended to run in real time, processing each document as it is seen. In addition, the TDT systems focus on organizing information over time and detecting novel information, rather than handling specific queries input by a user. Many IR systems for a variety of applications have been developed using the TDT framework and data sets. Here, I discuss three directions in TDT research in more detail - First Story Detection, timeline generation and update summarization.

First Story Detection

A 1999 summer workshop entitled "Topic-based Novelty Detection," held at Johns Hopkins University's Center for Language and Speech Processing, extensively studied First Story Detection [6]. In addition, it also aimed to address the New Information Detection task, which is the sentence-level analog (that will be discussed in Section 2.2.4). FSD addresses novelty at the document level, as the goal is to identify, in a stream of intermixed broadcast and newswire stories encountered one by one over time, those stories which discuss something novel - an event not related to a previously discussed topic. In particular, FSD systems must mark each arriving news story with a confidence score (between 0 and 1) that it is new, after the story is seen but before the next story arrives. At the workshop, various approaches to FSD were trained and tested on data from TDT-2. The corpus contained approximately 60,000 news stories, each of which was tagged as being on-topic or off-topic with respect to each of 96 news topics. The workshop team applied several approaches to FSD including the Vector-Space model, a technique that used named entities to identify new stories and a probabilistic model.

In the familiar Vector-Space approach, each story was represented as a vector of terms, with coordinates representing the frequency of a given term in that story. For the similarity function, the familiar Cosine Similarity metric was used. Next, two models for comparing each incoming document to the previously seen information were examined, namely agglomerative clustering and nearest neighbor clustering. While they found that agglomerative clustering does well in constructing cohesive clusters of topically-related stories, they found that nearest neighbor was better for FSD. The reason is that each story classified as novel should be sufficiently different from every previously seen story.

Based on the intuition that topically related documents should discuss a particular event involving the same people, locations and times, Allan and colleagues experimented with using named entities in detecting novel stories. Their corpus was tagged for seven types of named entities (person, organization, location, date, time, money and percent) and were not normalized (e.g. different names for the same person were not mapped to a single token). When features involving the presence of named entities were used in FSD, small improvements were noted. However, the researchers pointed out several reasons why this approach is not more helpful in the news domain. For example, in news reporting new entities are often introduced over time, even when the same topic is discussed. In addition, news reports are not always focused around particular people or organizations.

Another framework with which they experimented was the probabilistic approach to FSD. This involves, for each incoming document, finding the probability of a new topic occurring, given the current incoming story:

$$P(new|s) = \frac{P(s|new)P(new)}{\sum_{t \subset T} P(s|t)P(t)}$$

where T is the union of all topics that have been previously seen. In this approach, they developed a topic clustering method (t), language models, and a topic prediction model ($P(t)$). In addition, it is assumed that each story is generated by (the language model of) a unique topic, t. For the language models ($P(s|new)$ and $P(s|t)$), unigram distributions were deemed to be sufficient since the goal is to compare probabilities between topics rather than to predict likely sequences of words (as in the case of speech recognition). For topic prediction ($P(t)$), they used a geometric decay model, rather than simply using the relative frequency of each seen topic up to the present time, in order to take account of the fact that once a topic occurs in the news, it is likely to be discussed again within a short window of time. However, as time goes on it is less like to appear again. The performance of the probabilistic system was noted to be equivalent in performance to the best Vector-State model.

In short, in FSD systems, an incoming news article is compared to all previously seen documents, in order to determine if it discusses a new topic or story or is related to previously seen topic or topics. While a number of different approaches were applied and tested, none outperformed the best traditional, cosine-based vector system. The authors state that they do not expect to be able to improve their results “in absence of substantial changes in approach.”

	f_j	$f_{\bar{j}}$
f_h	a	b
$f_{\bar{h}}$	c	d

Figure 2.1: Counts needed for the calculation of the χ^2 statistic.

Timeline generation

Also using the TDT data and framework, [118] developed a statistical model of feature occurrence over time, in order to automatically generate overview timelines that describe the contents of the input corpus of texts. The output of their system is a ranked (by importance) list of groups of features that correspond to events or topics discussed in the news stories at each point in time. In contrast to traditional TDT systems, their system works retrospectively rather than in real time. Nonetheless, it relates to information detection over time in that in order to construct a timeline of events, novel news topics that are discussed must be detected.

Their model is based on classical hypothesis testing, more specifically, the χ^2 test of independence for the features observed. They assume that there is no association between the appearance of a pair of features in a given document. In this case, the observed features are named entities and noun phrases that have been tagged in the corpus. The statistic for discrete events used is the number of documents containing the feature during a particular time interval. It is assumed that the process is stationary (i.e. the probability of seeing a particular feature does not change with time) and that the processes generating any pair of features are independent.

Table 2.1 shows the information needed to calculate the strength of association between a given pair of features. For each feature, f_j and f_h, one needs the number of documents in which both features are present (a), the number in which f_j is not present but f_h is (b), the number in which f_j is present but f_h is not (c), and the number in which neither feature is found (d). Where N is the total number of documents in the time span under consideration, χ^2 is found as follows:

$$\chi^2 = \frac{N(ad - bc)^2}{(a + b)(a + c)(b + c)(b + d)}$$

In processing the documents, Swan and Allan's system first builds inverted lists for the named entities and noun phrases extracted from the documents. Next, the corpus is divided into days and finds the number of documents having each feature on each day. The χ^2 statistic is then calculated for each feature on each day. If the χ^2 value for a feature is above a set threshold for consecutive days, these days are combined in order to create a single time range. In order to get an idea of how distinctive a feature was at its peak, the χ^2 values are calculated for each subrange of the given time range, and the highest value is found. Then, once all the terms with significant appearances in the news documents have been identified along with their associated time ranges, they are sorted on their maximum χ^2 values. This results in a sorted list of key features that appeared in the news corpus and their respective dates of occurrence.

In order to identify TDT style topics, the features are then clustered with respect to their time ranges. Beginning with the highest ranking unclustered feature, if its time range overlaps with those of a given cluster, the χ^2 statistic is calculated for the cluster, including the candidate feature. If the value exceeds a threshold, the feature is marked as a potential member of the cluster. (Threshold values were trained and were different for named entity and noun phrase features.) Once the list of features has been processed (the initial clustering phase) average link clustering is performed on the marked candidate features in order to generate the final topic clusters. For each cluster, they automatically generated a topic name by assigning the highest ranked noun phrase as well as the highest ranked entity name. An example of the TDT-2 story topics identified by the system is shown in Figure 2.2.

Manual TDT label	**System assigned label**	**Date range**
Barry Goldwater dies	barry goldwater senate barry goldwater	May 29 - May 31
Daimler-Benz / Chrysler Merger	daimler-benz industrial merge	May 6 - May 10

Figure 2.2: Sample output of Swan and Allan's system.

An evaluation was conducted in order to assess how well the automatically generated clusters of terms corresponded to the official TDT topics (assigned manually by four judges). In addition, the automatically generated topic names were evaluated. The pairwise overlap of TDT topic and automatically generated cluster matches was found to be 86.7%, with good agreement between the four judges. However, the automatically generated cluster labels were not seen as being very helpful by the judges. Overall, however, the model represents a relatively simple way to identify topics in a stream of text documents.

Of particular interest to the tracking of specific information in text, the authors noted that while the method works well for stories that appear for short periods of time in the news, it was not able to identify topics that were long running. Since such long running stories may disappear and reappear in the news over time, with their features often changing, the system may recognize such stories as a series of separate events rather than one long running event.

Update summarization (δ-summarization)

Allan and his colleagues [4] also used the TDT data to develop a system to assist readers in monitoring changes in a stream of topically related news documents. In particular, they proposed the notion of update summaries (δ-summaries) that are produced over time and indicate only what has changed in the story. In other words, at each point in time when a δ-summary is produced, the user sees the new information only. Like the other TDT-related systems above, it is not a context-sensitive system. Rather, its goal is to keep readers informed of new information over time. The temporal summaries problem is as follows:

- Each news topics has a set of events and each sentence may discuss one or more such events.
- Sentences can be classified as being "on-event" or "off-event" in relation to each event. Some sentence may not be relevant to any event.
- The summarization system assigns a score (reflecting perceived importance) to each sentence and all sentences that are published at the same time t are considered for inclusion in the summary.
- The summary created at time t will contain all sentences with scores exceeding some threshold, θ.

In developing a system for producing δ-summaries, the authors were concerned with the precision and recall in identifying useful ("on-event") as well as novel sentences (defined as those describing events not previously covered up to the present time). The authors also make the assumption that novelty and usefulness of sentences are independent, such that for a given sentence, $P(useful \bigcap novel) = P(useful) * P(novel)$. This was so that a language modeling approach could be used. In this approach, given a text and a language model (LM) for each topic, one estimates how likely it is that the text was generated from that particular LM. Given a set of events in a particular topic, $E = e_1, e_2, ...e_v$, and a set of sentences, $S = s_1, s_2, ..., s_n$, the language model estimates how likely it is that the word w appears in that topic:

$$P(w) = \frac{\sum_i tf(w, s_i)}{\sum_i |s_i|}$$

In creating and evaluating their system, TDT2 data was used, in which each document was labeled as being "on-topic" or "off-topic," with respect to the list of 22 TDT topics. Human judges then established a list of events expressed in each topic. Two LM approaches were tested for measuring both novelty and usefulness, and are summarized in Table 2.3.

Measure	**Description**	**Formula**
$Useful_1$	Given a sentence, s_k and the LM for a given topic, p, this measures the likelihood that the sentence was generated by the topic LM. The LM is built from all sentences seen to date.	$P(useful_1) = P(s_k \| LM_p(s_1, ..., s_{k-1}))$
$Useful_2$	LM is now built from set S of all sentences in the story.	$P(useful_2) = P(s_k \| LM_p(S))$
$Novel_1$	Given a sentence s_k, this estimates the probability that two sentences (s_k and a previously seen sentence) could arise from the same LM.	$P(novel_1) = P(e(s_k) \neq e(s_i), foralli < k)$
$Novel_2$	Corrects for sparse data by grouping together sentences of the same event. Assumes that when sentence s_k arrives, there are m event clusters, c_1 through c_m.	$P(novel_2) = P(e(s_k) \neq e(c_i), foralli \leq m)$

Table 2.3: Allan and colleague's measures for estimating the usefulness and novelty of a given sentence.

Each of the usefulness and novelty measures was evaluated for retrieval of useful or novel sentences, respectively. Finally, the novelty and usefulness measures were combined, in order to create a single measure of the "interestingness" of an input sentence. As mentioned previously, in one implementation, Allan and colleagues assumed that the two qualities are independent, such that $P(interesting) = P(useful) * P(novel)$. In addition, they experimented with combining these two factors linearly, such that $P(interesting) = \alpha * P(useful) + (1 - \alpha) * P(novel)$. The methods were evaluated as to their average precision over many different summary compression rates and compared to baselines including natural order, round robin and random ordering. While the Useful-1*Novel-1 measure was the best in terms of average precision, outperforming all of the baselines, it was noted that the round robin baseline was quite competitive with the new methods.

In summary, Allan and colleagues, in working with the TDT framework of finding new information in documents encountered over time, developed new measures for retrieving useful sentences, novel sentences and "interesting" sentences, which are both useful and novel. This work is also highly related to that of Novelty systems, to be discussed in Section 2.2.4. Of note is that since the methods presented above use language models, which quantify the likelihood of seeing certain lexical items in a sentence from a given event, the approach is not robust to paraphrasing. In other words, if a sentence containing similar or the same information as a previously seen sentence, but expressed it using different words, it could still be identified as an interesting (relevant and novel) sentence. Therefore, it seems inevitable that update summaries will contain some amount of redundant information.

2.2.3 Information Filtering

While tasks related to TDT are event-based, information filtering, which involves monitoring document streams in order to find relevant items (with respect to a user's predefined profile), is subject-oriented [16]. In filtering, users create profiles that represent their information needs, in an effort to find items related to a given subject, as they become available over time. While traditional information filtering systems are Boolean, classifying each document in a stream as being either relevant or not relevant to a user's profile, more recent systems have incorporated the notion of novelty [129], in an effort to detect information that is not only relevant to a user, but that has not already been seen.

Noting that the nature of redundancy and novelty depends on what the user has already seen, the authors make the following three assumptions in developing their system:

- The redundancy of a given document encountered at time t, d_t, depends on all documents that the user has seen before, $D(t)$. If $R(d_t)$ is the measure of redundancy of d_t, $R(d_t) = R(d_t|D(t))$.
- How redundant d_t is depends on the set of relevant documents, $DR(t)$, that the user has seen up until time t, so $R(d_t|D(t)) = R(d_t|DR(t))$.
- For two document sets A and B, if $B \subset A$, and B makes d_t redundant, then A must also make d_t redundant.

They point out that not only are document timestamps important for determining what documents have already been seen at a given time t, documents are also more likely to be similar to others that are delivered around the same time. Another important point is that redundancy is not symmetric. For example, if one sentence is shown to the user at time t, and then an entire paragraph that contains the original sentence as well as others is shown at time $t+1$, the paragraph would most likely not be redundant. However, showing the paragraph first and then the sentence would certainly cause the sentence to be redundant.

Four different measures of document redundancy were proposed and evaluated:

1 *Set difference*: Each document is represented as a "bag of words," and the novelty of d_t is measured as the number of new words in a smoothed set representation of d_t. Its representation is smoothed in order to compensate for stop words that are common in the overall corpus as well as those that are common

in all documents on a given topic (topic stop words). If d_i is a document that has been previously seen,

$$R(d_t|d_i) = ||Set(d_t) \bigcap \bar{Set}(d_i)||.$$

2 *Geometric distance*: Each document is represented as a vector using each unique word as one dimension, such that

$$R(d_t|d_i) = cos(d_t, d_i).$$

3 *Distributional similarity*: This is a language modeling approach, in which document d_t is represented by its unigram word distribution, θ_d. The Kullback-Leiber divergence is used to measure the redundancy of one document given another, such that

$$R(d_t|d_i) = KL(\theta_{dt}, \theta_{di}) = -\sum w_i P(w_i|\theta_{dt}) \log \frac{P(w_i|\theta_{di})}{P(w_i|\theta_{dt}}.$$

4 *Mixture models*: In this approach, it is assumed that each document that is relevant to a user's profile is generated by a mixture of three language models - a general English model, a topic-specific model and a document-specific, "new information" model, θ_{d-core}, which can be estimated from training data. In this case, the measure of document redundancy is given by

$$R(d_t|d_i) = KL(\theta_{d_t-core}, \theta_{d_i-core}).$$

Once a measure of redundancy is implemented, a redundancy threshold is required. Zhang and colleagues estimated the user's tolerance for redundancy,

$$P[userjfindsd_tredundant|R(d_t|DR(t))],$$

from the training data. Their method for doing this involved adaptive feedback. Initially, the threshold is set so high that only extremely redundant documents are classified as such. When a document d_t is delivered to the user, he or she is then asked for feedback. If the user finds the document redundant and if $R(d_t) > R(d_i)$ then for all d_i, the new threshold is set to $R(d_t)$. Otherwise, the threshold is decreased to

$$thresh = thresh - \frac{thresh - R(d_t)}{10}.$$

The four measures of redundancy were evaluated using AP News and Wall Street Journal data from TRECs 1, 2 and 3. In collecting redundancy judgments, annotators were asked to mark each document as being "not redundant," "somewhat redundant" or "absolutely redundant." The authors considered the cases where both somewhat and absolutely redundant documents were treated as redundant, as well as just the ones marked as absolutely redundant. In both cases, the redundancy measures that performed the best (in terms of precision, recall and accuracy) were the cosine distance and the mixture model (language model). The fact that the cosine metric performed well surprised the authors, as they noted that redundancy is not symmetric between a given pair of documents, however, the cosine metric is symmetric. To contrast, the relatively good performance of the mixture model metric was not surprising, since it explicitly aims to model lexical items that are new in a given document.

2.2.4 Novelty systems

The goal of novelty systems is to identify information that is not only relevant but has also never been seen before by the user. While novelty has been incorporated into other tasks such as information filtering, as discussed above, in recent years there has been interest in attempting novelty detection at finer levels of granularity and in particular, at the sentence level. A means to classify a sentence, in a stream of documents over time, as being either relevant or not and, if relevant, novel or not, would be of direct benefit to a number of IR applications such as text summarization and question answering. This is because, as noted previously by [23], once a certain amount of information has been seen, additional sentences are likely to contain redundant information, such that one needs to carefully consider the balancing of relevance and redundancy.

Two research initiatives in particular have attempted to develop systems for the detection of novelty sentences - a summer workshop on topic-based novelty detection [6] and the TREC Novelty Track evaluation [49, 116]. Both of these efforts have indicated that novelty systems are difficult to develop, because what "novelty" constitutes is not easy to define and implement in an IR system. Therefore, this section focuses on discussing this issue, as it has serious implications for how systems that find changing information over time at fine levels of granularity can and should be built.

New Information Detection

At the 1999 Topic-based Novelty Detection workshop sentence-level novelty detection was addressed in addition to the First Story Detection problem. In comparison to FSD, the New Information Detection task was designed to operate within news stories rather than across them, as in my problem of interest. However, in their final report [6], the participants noted that little progress was made towards the sentence-level task. The main problem they cited for this is the fact that the meaning of "novel information" is very difficult to define and is quite context-dependent. Therefore, one question of interest is whether query-sensitive new information detection systems could be developed in future work.

TREC Novelty Track

A second major effort towards the development of sentence-level novelty detection systems is the TREC Novelty Track, which began in 2002[5]. In this evaluation, participants are presented with clusters of multiple, topically-related documents as well as TREC-style topic queries. These queries represent a general topic of interest to a user. Figure 2.3 shows an example of the title and description fields for a query for a data cluster from the TREC Novelty 2003 test data [116]. The goal is to build a system that, in processing the documents in chronological order by publication timestamp, can first identify the set of sentences that are relevant to the given topic. In a second step, novel sentences, which must be a subset of the relevant sentences, are also identified. The definition of novel information is "previously unseen information."

In the 2002 Novelty Track, several problems were noted with respect to the annotation of the truth data for the evaluation [49]. For each of 50 TREC topics, two judges were given the topic query and a set of relevant documents (presented to them in rank order in terms of relevance), and were asked to read though the set of documents, making a list of the *sentences* that were relevant to the topic. After that, they were to review their relevant sentences in order, and to eliminate those that did not contain novel information. One problem was that there were very few relevant sentences chosen, resulting in many negative and few positive relevance examples available to train the automatic novelty systems. At the same time, most of the relevant sentences were also marked as being novel. In addition, a major assessor effect was noted. The assessors typically did not choose the same proportion of relevant and novel sentences from the set, nor did they tend to choose the same sentences.

[5] $http://trec.nist.gov$

```
Title: Russian submarine Kursk sinks

Description:
Reports on what was known about the sinking of the Russian nuclear
powered submarine, Kursk, are relevant.  Speculation about what caused
the explosions aboard; description of the vessel and its capabilities,
and mention of efforts to rescue the crew are relevant. Reports
that U.S. submarines were monitoring Russian navy exercises and Russia's
suspicions that the Soviet submarine K-128 was struck by an American
submarine and sunk in 1968 are relevant. Mention of the fact that Russia
turned down a U.S. offer to send a deep-diving rescue vessel is relevant.
Discussion of U.S. plans to retire one of its two rescue vessels is not
relevant.  Polls reporting how Russians felt about the disaster and
mention of ceremonies for the dead are relevant.
```

Figure 2.3: Example TREC Novelty track query.

Because of these problems, in evaluating the novelty systems, comparisons were made using several different truth data sets. For example, systems were evaluated against the minimum set, or the set of sentences from the assessor who marked the fewest sentences as being relevant or novel.

In order to address the problems with the 2002 data, several changes were made to the annotation task in 2003 [116]. Two assessors were again used in the annotations, however, this time one was considered to be the official judge. The second assessor was used only for assessing the level of agreement on the task (not for evaluating the performance of the novelty systems). Other changes were that for each cluster, the official judge was instructed to search a database for the most 25 relevant documents and then, for the annotations, the documents were presented to the judges in chronological order (rather than ordered by relevance). The process for annotating relevant and then novel sentences remained the same.

The changes made in the 2003 task resulted in an improved novelty data set in several ways. First, the proportion of sentences marked as being relevant was much greater than in 2002. In addition, the percentage of relevant sentences that were also marked as being novel was lower. However, there was still a large assessor effect noted. While the interjudge agreement was not quantified, it was noted that while the judges, in general, tended to pick approximately the same numbers of relevant and novel sentences, they did not tend to pick the same sentences. Thus, the novelty annotation task, in the context of a general information query, appears to be annotator-dependent and not reproducible. This is an issue for the creation of a data set on which novelty systems can be trained and tested.

One group participating in the Novelty 2002 evaluation created additional training data by hiring their own annotators, but used TREC data clusters as well as the Novelty annotation instructions [68]. They also noted relatively low interjudge agreement, but noted that agreement was somewhat topic-dependent. In addition, Allan and colleagues have noted that of the two-part task of identifying relevant and then novel sentences, with respect to a query, the former appears to be the more difficult [7]. Citing the fact that by definition, novel sentences are a subset of relevant sentences, the performance of novelty detection systems quickly degrades as the accuracy of relevance detection is lower.

I am interested in building a system for tracking changing information over time, which can also be interpreted as following novelty over time. In contrast to the TREC Novelty setting, which focuses on novelty at the sentence level and in a general context, the proposed system will track specific information, stated as a factual question. It remains to be seen if a context-specific definition of novelty can be developed, that could result in more agreement in judgments between human assessors. In addition, as mentioned by [7] and [68], agreement on relevance judgments and performance of relevance recognition systems needs to be improved before additional progress can be made in the detection of novel sentences. Since very high levels of interjudge agreement for annotating facts in news texts has previously been achieved in other studies (e.g. [124]), using a query-sensitive question-focused framework might result in better agreement between judges in finding relevant and novel sentences in multi-document clusters of documents published over time.

2.3 Detecting Dissimilarity at Different levels of Granularity

In contrast to Section 2.2, which focused on systems specifically designed to track information change over time, in this section I will discuss more generally, techniques for detecting dissimilarity between textual units in tasks or systems that do not involve a temporal element. Obviously, this is a broad topic, which has been addressed in the context of many different IR and NLP tasks. However, the focus here will be on discussing the various techniques that have been applied to the problem of distinguishing similar and dissimilar information, at different levels of textual granularity. Table 2.4 summarizes the techniques discussed in Sections 2.2 and 2.3.

Approach	**Task**	**Granularity**
Vector-Space model (cosine similarity)	FSD	Document
Probabilistic language modeling	FSD, Novelty filtering	Document
Fingerprinting (probabilistic)	Version detection	Document
Identity measures (ranking)	Version detection	Document
Hypothesis testing with named entity features	Timeline generation	Document
Naive Bayes class. using (Ngram overlap, greedy string tiling and sentence alignment as features	Text reuse	Document
Logistic class. using semantic/syntactic features	SimFinder (finding similar paragraphs)	Paragraph
Vector-Space model	Relevance/novelty detection	Sentence
Probabilistic language modeling	Temporal summarization, Novelty detection	Sentence
Multiple sequence alignment	Paraphrase detection, generation	Sentence

Figure 2.4: Methods for comparing textual units at different levels of granularity.

```
Original text published by news agency:

A drink-driver who ran into the Queen Mother’s official
Daimler was fined 700 pounds and banned from driving
for two years.

Tabloid rewrite:

A DRUNK driver who ploughed into the Queen Mother’s limo
was fined 700 pounds and banned for two years yesterday.
```

Figure 2.5: Example of text reuse by a tabloid.

2.3.1 The document level

Text reuse

Clough [29] investigated techniques for detecting similarity of a given pair of documents in the context of journalistic reuse of text. Similar to Jing and McKeown’s work on cut and paste-based text summarization [56], they noted that journalists often apply a set of cut and paste operations to a newswire text, in “reusing” this text to create a new article for publishing. This is illustrated by the example in Figure 2.5 taken from [29].

In later work, Clough and colleagues [30] focused on developing algorithms for classifying news articles as being either wholly, partially or not derived from newswire sources. To this end, they built the METER corpus, which contains newspaper texts from nine British news agencies as well as newswire texts published by the UK Press Association on the same stories. Each news article was manually classified into one of the three categories (wholly-derived, partially-derived or not derived from newswire). Three approaches were used to measure text similarity:

- Ngram overlap: Given a source (copy) text A and a possibly derived news article B represented by the sets of ngrams $S_n(A)$ and $S_n(B)$, the proportion of ngrams in both B and A, the ngram containment $C_n(A, B)$ is

$$C_n(A,B) = \frac{|S_n(A) \bigcap S_n(B)|}{|S_n(B)|}$$

- Greedy string-tiling: Expresses the extent to which the strings of lexical items in the source text A can be used to "cover" those in the news article B. Given a maximal length of substrings ("tiles") to consider, the similarity between A and B is

  $$gstsim(A,B) = \frac{\sum_{i \subset tiles} length_i}{|B|}$$

- Sentence alignment: Each sentence in the candidate derived text, DT is compared to each sentence in the source document, ST to find a set of best matches. Given a DT and its set of matches from ST, three measures are computed: SNG is the sum of the lengths of the maximum length not overlapping shared n-grams with a length of 2 or more; SWD is the number of matching words sharing stems not in an n-gram counted previously in SNG; SUB is the number of substitutable terms (synonyms) not counted in SNG or SWD. Letting L_1 be the length of DT and L_2 be the length of the a given best match from the source text ST, the three scores in Table 2.4 are computed. The final similarity score, which ranges from 0 to 1 in reflecting the proportion of aligned sentences in the newspaper text, is a weighted interpolation of the three scores previously mentioned:

  $$sasim(A,B) = \delta_1 * PSD + \delta_2 * PS + \delta_3 * PSNG$$

  where $\delta_1 + \delta_2 + \delta_3 = 1$, and were empirically estimated to be 0.85, 0.05 and 0.1, respectively.

The authors built Naive Bayes classifiers using the three similarity measures as features, in order to predict the level of derivation of a given news article to a given

Measure	Meaning	Formula
PSD	Proportion of DT that is shared material	$\frac{SWD+SNG+SUB}{L_1}$
PS	Proportion of shared terms in DT and ST	$\frac{2*(SWD+SNG+SUB)}{L_1+L_2}$
PSNG	Proportion of matching ngrams in DT and ST	$\frac{SNG}{SWD+SNG+SUB}$

Table 2.4: Measures used in Clough's sentence alignment approach to similarity.

copy text. They also experimented with using different combinations of their features in the classifiers. While all classifiers beat the baseline, the one combining all three features performed the best. The classifier performs best on the wholly-derived texts, and the authors note that such texts can be identified with $> 80\%$ accuracy.

While similar in spirit to other document-level tasks, such as TDT First Story Detection, in that it aims to classify a given document as being significantly different ("novel" or "not reused") from another or not, Clough and colleagues employed different similarity metrics than the classic cosine similarity or language model approaches that are often used in information retrieval. However, as noted by the authors, the metrics tested, like those discussed previously in Section 2.2, are based on lexical similarity and cannot sufficiently detect similarity between two texts. Therefore, they noted that improvements to their classifier will only be realized if more advanced NLP techniques are utilized.

Document versioning

Similar to the work of Clough and colleagues, [53] endeavored to develop a means to detect documents that are coderivatives of one another such as versions of the same evolving document. They noted several properties of coderviatives including the presence of the same rare misspelled words, common grammatical errors and unusual usages of words. In the current work, they experimented with multiple implementations of two general approaches, ranking and fingerprinting. In ranking, a common information retrieval technique, one attempts to produce a ranked list of potential coderivative documents. To contrast, in the fingerprinting approach one creates a representation ("fingerprint") for each document, and then compares the fingerprints of documents in order to determine how similar they are. In [53], they address the one-to-n problem, such that they compare a single document query to an entire collection. This is in contrast to the n-to-n problem in which every pair of documents is compared.

For the experiments with the ranking technique, four different similarity measures were used to create ranked lists of potential coderivatives for a given query and were then evaluated. Given the following definitions and notions,

- N: number of documents in the collection
- n: number of distinct terms in the collection
- f_t: number of documents that contain term t
- $f_{d,t}$: number of times term t appeared in document d
- f_d: number of total terms in document d
- W_d: weight (length) of document d
- D: the document collection
- q: the query document
- d: a document in collection D

these approaches are summarized in Table 2.5. In each case, "query terms" are produced from the given document for which one wants to find potential coderivative documents. Two types of similarity measures were implemented and evaluated;

the first three are standard IR measures while the last one, which has five different variations, is a new measure. While the standard measures are intended for ad hoc querying, the new identity measure is based on the assumption that coderivative documents should have similar *numbers of occurrences* of words, in addition to sharing similar words.

Metric	**Description**	**Formula**
Inner product	Gives a high weight to documents in which query terms appear frequently.	$\sum_{t \subset q \bigcap d} (1 + \log_e f_{d,t}) * \log_e(1 + \frac{N}{f_t})$
Normalized inner product	Normalized version of the inner product that addresses the problem of long documents being favored.	$\frac{1}{\sqrt{f_d}} * \sum_{t \subset q \bigcap d} (1 + \log_e f_{d,t}) * \log_e(1 + \frac{N}{f_t})$
Cosine measure	Attempts to compensate for differences in length by normalizing the inner product for document weight.	$\frac{1}{W_d} \sum_{t \subset q \bigcap d} (1 + \log_e f_{d,t}) * \log_e(1 + \frac{N}{f_t})$
Identity 1 (I1)	Makes use of the term weight differences between term frequencies in query and document, and document lengths.	$\frac{1}{1+\|f_d - f_q\|} \sum_{t \subset q \bigcap d} \frac{log_e \frac{N}{f_t}}{1+\|f_{d,t} - f_{q,t}\|}$
Identity 2 (I2)	Uses the log of the differences in document lengths as a discriminator such that the measure is not as sensitive to this difference as in I1.	$\frac{1}{1+\log_e(1+\|f_d - f_q\|)} \sum_{t \subset q \bigcap d} \frac{log_e(1+\frac{N}{f_t})}{1+\|f_{d,t} - f_{q,t}\|}$
Identity 3 (I3)	Gives more weight to documents having rare terms in common with the query.	$\frac{1}{1+\log_e(1+\|f_d - f_q\|)} \sum_{t \subset q \bigcap d} \frac{log_e(1+\frac{N}{f_t}*(f_{d,t}+f_{q,t}))}{1+\|f_{d,t} - f_{q,t}\|}$
Identity 4 (I4)	A slight variation of I2, it uses a different term weight discriminator.	$\frac{1}{1+\log_e(1+\|f_d - f_q\|)} \sum_{t \subset q \bigcap d} \frac{log_e(\frac{N}{f_t})}{1+\|f_{d,t} - f_{q,t}\|}$
Identity 5 (I5)	Emphasis on the term weight is increased such that rare terms have a much larger weight than common ones.	$\frac{1}{1+\log_e(1+\|f_d - f_q\|)} * \sum_{t \subset q \bigcap d} \frac{\frac{N}{f_t}}{1+\|f_{d,t} - f_{q,t}\|}$

Table 2.5: Hoad and Zobel's similarity metrics used in creating ranked lists of potential derivative documents.

In contrast to the ranking approaches using the similarity measures in Table 2.5 that use term frequencies, a compact description of each document is produced in the fingerprinting approach. Fingerprints are then compared in order to estimate the probabilities of documents being coderivatives. Each fingerprint consists of a set of minutiae, integers representing the document. Substrings of text are selected from each document, and a mathematical formula must be applied in order to calculate each minutia. Finally, the number of minutia in common between the query and each document determines the document's score. Given this setup, Hoad and Zobel note that there are four areas of considering for developing a fingerprinting procedure:

- The choice of function used to generate minutiae. In the present work, a hash function is used.
- The size or granularity of the substrings to be extracted. While using too fine a granularity can mean that the fingerprint becomes too susceptible to false matches, if the substrings are too coarse, the fingerprint may be too sensitive to change.
- The number of minutiae used.
- The choice of the algorithm to extract substrings.

In the fingerprinting experiments, the authors tried different anchoring mechanisms for the selection of the substrings, such as structure-based selection (e.g. starting at the k-th word in the sentence or paragraph), frequency-based selection (e.g. anchoring at the rarest words or prefixes) and positional selection (e.g. taking the first r words). They also varied the granularity of the substrings to be fingerprinted from a size of 1 to 20 words.

Finally, in evaluating their methods, the authors computed precision and recall on a number of runs, as well as two new metrics that they introduced:

- Highest false match (HFM): the highest score given to an incorrect result.

- Separation: the lowest correct result minus the highest false match.

Therefore, a good method would result in either a high HFM and high separation or both a low HFM and separation. The new metrics give credit for ranking the correct documents ahead of other documents, which recall and precision do not do. In their evaluations, the best method overall was the ranking approach using the Identity 5 similarity measure, which had an HFM of 11% and a separation of 25%. In comparison, the popular cosine similarity metric achieved an HFM and separation of 19% and 49%, respectively. Thus, for the task of detecting document versioning, the newly proposed measure performs better than the standard IR measures.

2.3.2 The sub-document/paragraph level

In the context of a multi-document summarization project, in which natural language generation techniques were applied, [52] developed the SimFinder tool. SimFinder uses a machine-learned similarity measure at the paragraph level in order to classify whether or not two paragraphs contain "common information." Syntactic and semantic features of the input paragraphs were used in building a logistic classifier. For example, some of the semantic features employed include the matching of exact words, word stems and WordNet synonyms. In addition, syntactic relationships, such as subject-verb and verb-object pairs, were also used to make comparisons. Source information was also used as a feature in the model, as it was hypothesized that two very similar paragraphs would be unlikely to come from the same source article.

The logistic regression model converts the evidence from the features into a similarity metric, that ranges from 0 to 1. Using this metric, the paragraphs can then be clustered to form groups of topically related texts. The clusters are then used to create summaries in a variety of ways, however, the central idea is to select one paragraph or sentence from each cluster. Thus, when given an input set of topically-related documents to summarize, SimFinder helps prevent redundancy in the resulting summaries.

2.3.3 The sentence level

The sentence analog to the problem of detecting similar paragraphs is the problem of paraphrase identification. [72] presented an unsupervised learning approach to identifying inference rules from a corpus of news articles. In this work, the distributional hypothesis, that lexical items appearing in similar contexts tend to also be close in meaning, is extended to paths in dependency trees. The algorithm developed computes the similarity between two paths, in identifying semantically related pairs of inference rules such as “X resolves Y” and “Y is solved by X.” The algorithm was used to generate such rules for the first six questions in the TREC-8 Question Answering Track, and the paths were compared to a set of human created paraphrases. The authors obtained conservative yet promising results in a first attempt at automatically discovering inference rules in a large corpus of new texts. To contrast, [15] used a corpus of aligned texts that had been translated independently into English by various translators, in order to examine paraphrase extractions. Their algorithms used sentence features such as lexical descriptions and syntactic patterns, in order to determine whether or not a given pair of sentences were paraphrases.

In addition to paraphrase recognition, recent research has also focused on learning how to generate paraphrases of sentences. [14] presented an unsupervised method for producing multiple paraphrases given an input sentence. Using multiple-sequence alignment (MSA), the method learns to generate paraphrases using comparable corpora - unannotated news articles on the same topic collected from different news sources. First, sentences are clustered by topic. In order to allow for variability in arguments in sentences, all appearances of dates, numbers and proper names are replaced with generic tokens. Next, MSA is performed on the sentences in each cluster, using edit distances between each pair of sentences as the distance metric. The results of MSA are then represented as word lattices that show the structural similarities between the sentences in the cluster. Once lattices have been computed for each comparable corpus pair, lattice paraphrase pairs are identified as those that tend to take the same argument values. The word overlap between the set of argument values taken by two lattices is computed, with proper names and numbers receiving double weight. Two lattices are then paired if their overlap exceeds a tuned threshold. Finally, given an input sentence to paraphrase, it is first aligned with one of the lattices. If alignment is successful, one of its comparable corpus paraphrase lattices can then be used to rewrite the sentence.

Another approach to generating paraphrases used syntactic information in extracting Finite State Automata (FSA) or word lattices that can be used to generate paraphrases [90]. The input to the system is a group of sentences that correspond to the same meaning. For each sentence, a syntactic parse tree is produced. Next, parse trees of sentences with similar syntactic structures are merged top-down. For example, if two sentences expand into NP-VP elements, it is assumed that the NPs and VPs of the two sentences can be merged. Keyword checking is done in order to prevent erroneous alignments. For each node in the tree, a list of keywords that are spanned by the node is kept. Nodes from two trees are aligned only if they share common words in their keywords lists. This entire process is referred to a "mapping parse forests." Once the parse forests have been mapped, they are simply traversed in order to create the corresponding FSA. Alternative paths between any two nodes at the start and end of the FSA are assumed to be paraphrases of one another. The final step in the algorithm is to "squeeze" the FSA. Because of the strict criterion in the tree merging process, small differences in syntactic structure can prevent some legitimate mergings from occurring. In squeezing, if two edges going into or out of a node in the FSA have the same word, the nodes on the other end of the edges are merged.

2.4 Recognizing Semantic Relationships between Textual Units

The current section discusses work related to two types of semantic relationships, discourse and temporal relationships, that hold between two text spans. The automatic recognition of discourse relationships between two textual units has been put forward as a means to improve IR systems such as extractive single [77] and multi-document summarization [130]. However, it is also relevant to the proposed system, in which I aim to track factual questions over time. For example, in the case of an elaboration relationship between two sentences, it might not be sufficient to answer a user's question by returning only one of the two sentences. In addition, since we want to find information over time, temporal relationships between sentences need to be considered. Section 2.4.2 surveys previous work towards the development of methods for the automatic resolution of temporal relationships between events discussed in a given text.

2.4.1 Discourse relationships

Theories of textual structure and cohesion, typically coming from the linguistics and computational linguistics communities, attempt to describe the nature of written texts and how elements of a text fit together. Such theories are important to text understanding and have been used by several researchers in implementing

```
[Although Brooklyn College does not yet have a junior-year-abroad
program,] [a good number of students spend summers in Europe.]

Nucleus: a good number of students spend summers in Europe.

Satellite: Although Brooklyn College does not yet have a junior-year-abroad
program,

Rhetorical relation: contrast
```

Figure 2.6: Example of the RST relation "contrast."

computational models of discourse in texts. Here I discuss work that is relevant to understanding the relations that hold between topically-related texts written over time, from which the proposed IR system would aim to find answers to the user's input question.

Single document

Rhetorical Structure Theory (RST) has contributed a great deal to the understanding of the discourse of written documents [76]. RST describes the coherence nature of a text and is based on the assumption that the elementary textual units are non-overlapping text spans. The central concept of RST is the rhetorical relation, which indicates the relationship between two spans, a nucleus and its satellite. The core RST discourse relations are elaboration, contrast, exemplification and narrative sequence. For example, a case of elaboration would be where a text span (satellite) expands upon something that was introduced by another span (its nucleus). A simple example of the "contrast" relation (taken from [77]) is given in Figure 2.6.

RST has been used in sentence selection for single document summarization [77]. However, it cannot be applied to the analysis of multiple documents. In RST, text coherence is achieved because the writer intentionally establishes relationships between the phrases in a text in order to convey a desired message to the reader. This is not the case in the multiple document setting, where we may want to analyze a set of articles that are topically related but that have been written by different authors.

Multi-document

Inspired by Rhetorical Structure Theory, [103] endeavored to establish a Cross-document Structure Theory (CST) that is more appropriate in the multiple text setting. CST focuses on the relationships between sentences that come from different documents, which vary substantially from those between sentences in the same text. Figure 2.7 shows some examples of CST relationships.

Relationship	**Description**	**Text span (S1)**	**Text span 2 (S2)**
Equivalence (paraphrase)	S1 and S2 convey the same information	Derek Bell is experiencing a resurgence in his career.	Derek Bell is having a comeback year.
Subsumption	S1 contains all the information in S2, plus additional information not in S2.	With 3 wins this year, Green Bay has the best record in the NFL.	Green Bay has 3 wins this year.
Contradiction	S1 and S2 contain conflicting information.	There were 122 people on the downed plane.	126 people were aboard aboard the plane.
Overlap (partial equivalence)	S1 provides facts X and Y while S2 provides facts Y and Z; X, Y and Z are non-trivial.	The plane crashed into the 25th floor of the Pirelli building in downtown Milan.	A small tourist plane crashed into the tallest building in Milan.

Figure 2.7: Examples of 4 Cross-document Structure Theory relationships.

CST relationships characterize the similarities between cross-document sentences (e.g. paraphrase, when two sentences express the same concepts in different ways or partial overlap, where sentences overlap with respect to the information that they convey). They also express complementarity (e.g. historical background, when one sentence provides history of an event described in the other sentence). Finally, some CST relationships express differences between a pair of sentences (e.g. contrast, contradiction).

[130] showed that CST relationships can be used to improve the quality of extractive multi-document summaries. Specifically, it was shown that human judges prefer summaries in which more CST-related sentences were included, as compared to the default summaries produced by their summarizer. Finally, Zhang and colleagues have shown the feasibility of detecting CST relationships automatically [131] [132].

2.4.2 Temporal relationships

As noted by many researchers working with clusters of related documents (e.g. [75], [95] and [13]), readers must be able to determine when each event that is discussed happened in order to fully comprehend a text. However, events are not necessarily described in chronological order, particularly in narrative texts such as news stories [81]. Therefore, in order to develop a system for tracking changes in text over time, a system must be able to accurately resolve temporal relations in text - both on an absolute timeline as well as establishing the relative ordering between events described in a text. This is a challenging task since temporal relations are not always expressed as explicit times and dates of events, but rather, they are often indexical (e.g. two Thursdays ago). Further, a study by [107] found that authors often mean slightly different things by the same temporal expression. In an effort to learn the meaning of usage of time phrases for use in natural language generation systems, they studied weather reports written by five different authors. They aligned the texts with the numeric meteorological data that was used to write the reports, and extracted explicit times for each expression. They found that certain phrases, such as "by midday," tended to mean the same thing for all authors (12 noon, in this case). However, other phrases like "by evening" had more variation in meaning across authors. The remainder of the section will discuss some recent developments addressing the challenges of recognizing temporal relations in text.

Automatic timestamping

Several efforts have addressed the automatic "timestamping," or the resolving of the absolute time of events expressed in text, in addition to establishing the relative ordering of events described. Much of this work has focused on the domain of news. [75] developed a method for the temporal processing of news using manually constructed rules that were then augmented with machine learned rules. In addition to resolving explicit time expressions such as "Tuesday, November 5, 2000," they also focused on indexical expressions that express time in a relative fashion, such as "two weeks ago." Their system processes text that has been tagged for part-of-speech. In the first step, explicit, self-contained time expressions are identified and are represented in the ISO standard format (e.g. an expression such as "June 1999" is represented as 19:99:06). Next, a discourse module resolves the context dependent expressions using a set of ordered rules that aim to determine the direction and offset from the reference time. The rules attempt to find lexical markers that indicate offsets from reference times (e.g. "this coming Christmas" or "next month") and use nearby dates to infer a direction from the reference time. In evaluating the system performance, both print and broadcast news were used. In the test data set, time expressions were tagged and were assigned time values. The authors reported an accuracy (F-measure) of 83.2% as compared against the hand-tagged test data set.

To contrast, [112] developed a semantic tagging system for temporal expressions. In addition to recognizing core expressions as in Mani and Wilson's work, they also recognized the information conveyed by propositional phrases, such that phrases like "by Friday" were treated differently than the expression "Friday." In the first phase of processing, POS-tagged text is fed to a set of finite state transducers that are based on manually written rules. The FSTs extract temporal expressions based on syntactic

System	**Method(s)**	**Prec.**	**Recall**
Mani & Wilson (00)	manual and machine learned rules	83.7	82.7
Schilder & Habel (01)	FSTs based on manual rules	92.11 (simple) 87.30 (complex)	94.09 (simple) 90.66 (complex)
Filatova & Hovy (01)	manual rules	NA	82.29
Mani et al. (03)	machine learned rules based on clause features	NA	84.6

Table 2.6: Summary of time tagging systems' performances.

information. In proposing a meaning for each of the extracted expressions, they made a distinction between event-denoting expressions (verbs) and time-denoting expressions (prepositional phrases). Finally, related semantic attributes are linked in deriving the meaning of each temporal expression in a given sentence.

The authors evaluated their method for tagging the temporal expressions based on syntax against a corpus of manually labeled texts. In order to compare their results to those of Mani and Wilson, they reported precision and recall for both simple expressions and complex expressions (including the information contained in propositional phrases, that was not done previously).

The approach taken by [41] was to first break the sentences in each news story into separate event clauses and then to assign either timepoints or intervals to each clause. They analyzed both explicitly stated time expressions as well as implicit ones, indicated through verb tense. Once sentences were broken into event clauses, each clause was restated as a full sentence, such that pronouns were replaced by their antecedents. Date stamps occurring in the articles were also extracted. Next, each event clause is timestamped. They employed two sets of rules - one for clauses that contain explicit date information and another set for clauses without explicit date information. For cases where no explicit date is available, they first try to find either the day or the week or date of the month in the text that matches that in the document's timestamp. In addition, if an expression such as "X days ago" is present, they assign a resolved date or date range with respect to the article's reference time.

Finally, [74] developed a domain-independent machine learning approach to anchoring and ordering events in news. First, time expressions are tagged using the rules they previously developed in [75]. Next, clauses are tagged in the sentences. Since they found that only 25% of clauses in their data had explicit time expressions, it was not enough to anchor events to explicit times and thus, a reference time (tval) was computed for each clause. In particular, if an absolute time expression is stated in the clause, it is assigned. If a reporting verb is used, the document's timestamp is assigned. If the clause is a quote that is contained in the larger clause, j, the tval for j is assigned. Otherwise, if none of the above conditions applies, the tval assigned to the clause is the most recent one in the history.

A classifier was trained in order to establish the reference anchor, tval, for each clause. Given features of the clauses such as verb tense, paragraph and sentence number in the document and clause type (regular, complement or relative), the classifier predicted whether the last seen tval (from the previously seen clause) should be kept, if it should revert back to an earlier tval or if it should shift. Using the combination of the rules for the initial assignment of tvals plus the machine learned classification rules, the events in a test data set were assigned temporal anchors with 84.6% accuracy.

2.5 Challenges for Future Work

In this chapter, I have surveyed areas of previous work that are relevant to the building of an IR system for finding specific, dynamic information from Web documents. As discussed, several research areas are related to this problem including the detection of changing information over time, and recognizing semantic relationships between units of text. Perhaps the previous work most relevant to the system proposed in Chapter I is that of novelty systems [116], in that they attempt to recognize changing information at the finest level of granularity of any IR systems yet - the sentence level. However, as discussed, there is low interrater agreement on the task of finding sentences relevant to a topic of interest, and from the set of relevant sentences, identifying those containing previously unseen information. Therefore, the novelty framework needs to be refined.

In contrast to the novelty systems, my proposed system aims to identify answers to specific questions over time. Therefore, one question to answer in future work is if better interrater agreement might be achieved on the task of finding sentences that contain relevant and novel information with respect to a given factual question. In contrast to the previous results indicating that human judges do not agree on sentences relevant to a general topic, other studies have shown a high interrater agreement for annotating factoids and other semantic units in texts (e.g. [124]). If acceptable levels of agreement can be reached on finding both relevant and novel answers, then this would motivate the development of a means to automatically detect which answers are novel. However, if fact-based novelty judgments are too subjective, this would suggest taking a different approach to building the system. For example, rather than automatically classifying which extracted answers to a given question are novel and which have already been seen, the system could display all extracted answers to questions over time and across sources, in letting the user decide which ones are of interest to him or her.

Obviously, one other issue is whether or not to use the more lexically-based IR methods for building the system, or to try to incorporate some of the NLP techniques that analyze texts at a deeper level. As mentioned in several of the reviewed studies (e.g. [6, 30]), system accuracy can only be pushed so far using methods that consider only lexical techniques. At the same time, as noted by [66], the NLP techniques that involve deep syntactic or semantic parsing are slow and are currently not practical for use in IR applications designed to run on the Web. Therefore, as a starting point, I propose to attempt to build a system for query-specific, dynamic IR using components of two existing state-of-the-art systems, the MEAD summarizer [101] and the NSIR question answering system [99], and to evaluate its ability to help users find dynamic yet specific information from the Web. As discussed in Section 2.4.2, I expect to find challenges in relating the extracted answers to a point in time, since the extracted answers to a given question should not necessarily be mapped to the publication timestamp of the respective source document.

In conclusion, there is a need for IR systems that can help users find specific information in a dynamic, online environment. Such a system would combine the query-sensitivity of question answering systems with the notion of the time and source-dependency of answers, so that users could use the system to get a reliable, "big picture" of a changing situation such as a public emergency, from a number of Web-based sources. Certainly, the challenges in building such a system are many. The current review of the literature has suggested in particular, that a first step towards this endeavor should be the establishment of a context-specific framework for analyzing relevance and novelty at fine levels of textual granularity.

CHAPTER III

An Empirical Analysis of Dynamic Facts in Online News

In order to better understand how changing information is conveyed in a set of articles written over time, I built a corpus of breaking news stories that were manually annotated for important factual questions and their respective answers. I then conducted an empirical analysis of the corpus of question and answer sets with three goals in mind. As previously discussed in Chapter I, it takes time for the facts surrounding a breaking news story to settle to the point that all sources report the same information. While some facts change due to an ongoing investigation in the world, other reported facts change when news sources do not have accurate information that is later corrected. Therefore, the first goal of the empirical study was to characterize the reliability of information reported in the news stories in my corpus.

The second goal was to describe how answers to a given question change over time. Specifically, I used the Cross-Document Structure Theory semantic framework [103] in order to study how answers published over time relate semantically to the first answer published to a given question. Finally, I investigated the relationship between vocabulary usage in reporting a given fact, and publication time and source differences. This will be described in Section 3.4.

Story	**Source**	**Documents**	**Questions**	**Answers**	**Sample question**
Iraq suicide bombing	News Track	33	18	363	Who was the target of the attack?
Asian tsunami	News Track	146	5	40	Which countries were affected?
Milan plane crash	News Track	56	15	621	How many were injured?
RI nightclub fire	News Track	43	13	389	How many people were inside the building?
Columbia shuttle disaster	News Track	41	9	234	Where was debris found?
Gulfair plane crash	News Track	11	25	208	How many victims were there?
Kursk submarine disaster (N33)	TREC	25	20	211	Why did the Kursk sink?
Egyptair crash (N4)	TREC	25	22	265	Where did the plane crash?
China earthquake earthquake (N43)	TREC	25	8	106	What was the magnitude of the quake?

Table 3.1: Corpus of emergency news stories: story source, number of documents, questions and extracted answers, and a sample question.

3.1 Corpus

The corpus used in the empirical analysis consists of 9 multi-document clusters of breaking news stories describing emergency situations. I included two types of clusters in the corpus. “News Track” clusters were collected manually by tracking a predefined set of ten online news outlets. In particular, all articles published about the stories were tracked and downloaded for a period of 48 hours, in order to catch the new developments and updates. I also included three TREC Novelty clusters, which were taken from the Novelty Track 2003 test data [116]. The attributes of the document clusters are shown in Table 3.1.

In order to generate a collection of questions and answers for each news story, volunteer judges were recruited. For each story, one judge was asked to read through the articles and to come up with a list of factual questions that he or she deemed key to understanding the story. Between 15 and 30 unique questions were generated for each story. Next, I assigned each cluster to another judge, who was responsible for finding the answers to the questions for the respective story. More specifically, for each question, the judges went through every document in the cluster and found all answers to the question, listing the answer itself, as well as the document and sentence number where the answer was found. In the instructions, the judges were told to find only explicit answers. In other words, they were not to list sentences that merely provide information that allows one to infer an answer.

In a handful of cases, the judges found few answers to a given question. Since I was interested in studying how answers change with time, I eliminated the questions with fewer than three answers from the data set. In total, the corpus consists of 135 factual questions across the 9 news stories. The total number of annotated answers (to all questions) in the collection is 2,437. Once the answers were collected, one judge went through all sets of questions and answers in the corpus and indicated, for each answer set to a given question, if all the extracted answers expressed the same meaning or if the answer set contained some mutually exclusive answers.

3.2 Reliability of Reported Answers to Factual Questions

Using the 135 sets of questions and respective answers, I considered the following questions:

1. What is the prior probability of reporting a correct answer (at any point in time)?
2. What is the probability of observing a wrong answer?
3. Which source(s) tend to report the finalized answer first?
4. Which proportion of answers also state a primary attribution for the published

information?

3.2.1 Definitions

In answering these questions, I first needed to define the terms "correct answer," "wrong answer" and "finalized (or stabilized) answer." To illustrate these definitions, Figures 3.1 and 3.2 show answers to two questions in the corpus, from a small set of sentences previously discussed in Chapter I. Figure 3.1 shows four of the answers found concerning the question "How many were killed?" in the RI nightclub fire story. Similarly, Figure 3.2 shows four answers for the question "What was the plane's destination?" in the Milan plane crash story.

An example of a finalized answer is the fourth answer shown in Figure 3.1, "ninety-six people." As previously discussed, information in a breaking news story often takes time to settle and reach a ground truth. Incorrect or partially correct information may be reported and then updated or retracted over time as news agencies learn more about a situation. Therefore, I defined the term "finalized answer" to mean the answer upon which all news sources agree. In addition, once a finalized answer is reported, it does not change again. In the example, it can be seen that the fact "number killed" was most likely updated over time in order to reflect changing information from an ongoing investigation at the scene of the fire.

I defined the term "correct answer" to mean that the given answer reflected correct information at the time of publication and does not contradict the finalized answer. The "correct information" is defined to be that information reported by the majority of news sources at a given point in time. For instance, all of the answers shown in Figure 3.1 were considered correct, since they reflected the information reported by all news sources at each point in time. It is clear that the first three answers are not yet settled since they all express a minimum number of people killed (using the phrase “at least”), and that they do not contradict the finalized answer of “96.” However, it must be noted that a correct answer is not necessarily a finalized answer.

To contrast, an incorrect answer clearly contradicts the finalized answer. In other words, it expresses incorrect answer that must be retracted (rather than updated) at a later publication time. An example is given in Figure 3.2. Here, the incorrect answer, that the plane that had crashed was heading to Rome, Italy, was reported several times before the finalized information, that the plane was headed to the Milan airport, was reported.

```
Q: How many were killed?

02/21/03 01:03 (ABC News)
"at least 10 deaths"

02/21/03 06:41 (CNN)
"at least 26 people"

02/21/03 11:00 (MSNBC)
"at least 60 people"

02/21/03 21:45 (CNN)
"ninety-six people"
```

Figure 3.1: Answers to the question “How many were killed?” in the RI fire story.

```
Q: What was the plane’s destination?

04/18/02 13:17 (CNN)
"Rome, Italy"

04/18/02 13:42 (ABCNews)
"Italy’s capital, Rome"


04/18/02 13:42 (CNN)
"Rome, Italy"

04/18/02 13:42 (FoxNews)
"Milan’s Linate airport"
```

Figure 3.2: Answers to the question “What was the plane’s destination?” in the Milan plane crash story.

3.2.2 Findings

In addressing the four questions of interest, I manually examined all 135 question and answer sets that had been annotated by the judges. For each question in the corpus, I went through the extracted answers, associated with their respective publication source and time and in chronological order, and labeled them as whether they expressed correct or incorrect information and whether or not they expressed the finalized answer.

Over all of the 2,437 answers identified by the judges, 74.5% of them were correct answers, according to the definition explained previously. I concluded that the prior probability (e.g. knowing nothing about the source of information, publication time, or type of question and answer) of reporting a correct fact surrounding a breaking news story is approximately 0.75. Therefore, almost 25% of the answers expressed incorrect information. However, once the finalized answer had been reported at least once (by any given news source), I found that the probability of observing an incorrect answer drops to only 10.8%. Finally, only 14.6% of the answers in the corpus represented finalized answers, such that the majority (85.4%) of answers represented either incorrect answers or those that were still changing (i.e. "unsettled information") at the time they were reported. This statistic clearly illustrates the need for a means to handle dynamic question answering situations, such that users receive the most accurate information available.

In answering question three, it was not surprising that the larger Web-based news outlets tended to report the final answers to questions first. ABC News reported the greatest number of finalized answers in the corpus (26). Table 3.2 shows the top (and the worst) three news agencies, in terms of answering the specific 135 questions in my corpus.

Finally, regarding question four, only approximately one-third of the reported answers in the corpus (29.8%) stated a primary attribution. Here, whether or not the information is attributed to its source appears to depend on the question being asked. For example, for many questions that are more objective in nature (e.g. "From where was the plane coming?" "What was the weather at the time of the crash?") an attribution was typically not stated. To contrast, answers to questions about the cause of an incident or a situation (e.g. "What caused the fire?"), or that follow from an investigation (e.g. "How many people have been killed?") were more likely to state the source of the information.

Agency	**% of finalized answers reported first**
ABCNews	14.3
CNN	11.1
New York Times	9.5
CBC News	1.6
The Guardian	0.008
CBS News	0.008

Table 3.2: Best and worst news outlets.

3.3 Semantic Relationships between Answers

I used a subset of the Cross-document Structure Theory (CST) [103] to characterize the semantic relationships between the answers in the corpus. CST seeks to describe the relationship between a given pair of related sentences extracted from different documents. It proposes 18 relationships, which are not mutually exclusive. I used six of them, plus an additional relation, in designing a classification scheme for analyzing answers to a given question. In contrast to the original CST, in my scheme the relationships are mutually exclusive. Table 3.3 describes the relationships analyzed, including the new relation, "degree of certainty."

For each question in the corpus, I identified the finalized, or stable, answer. Then,

Relation	**Description**	**Question**	**A1**	**A2**
Identity	A1 and A2 are the same	What kind of aircraft was involved?	a Piper plane	a Piper plane
Paraphrase	A1 and A2 express the same information	What kind of aircraft was involved?	a Piper plane	a Piper aircraft
Contradiction	A1 and A2 are mutually exclusive answers	Who was on board the plane?	only the pilot	a pilot and co-pilot
Attribution	A2 is an attributed version of A1	What kind of aircraft was involved?	A Rockwell Commander	according to CNN, a Rockwell Commander
Elaboration	A2 is more detailed than A1	When did the crash happen?	during rush hour	at 5:54 p.m.
Partial Overlap	A1 provides facts X and Y; A2 provides Y and Z; X, Y and Z are non-trivial to the question	When did the crash happen?	on Monday, at 5:54 p.m.	April 18th at 5:54 p.m.
Degree of Certainty	With respect to a question of quantity, A1 provides a range of values while A2 provides one	How many were killed?	at least 13	13

Table 3.3: Relationship, description and an example question and answer pair, A1 and A2.

I manually classified all of the other answers to each question in relation to the final answer, using the scheme.

3.3.1 Analysis

The questions of interest to me were:

1. What proportion of answers are the same as (identical to) the finalized answer?
2. What proportion of answers are paraphrases of the finalized answer?
3. What is the distribution of CST types over all answers in the corpus?
4. Is there a pattern of CST relationships as answers stabilize over time?

Across all questions, 31% of the extracted answers were identical to the respective finalized answer, while 15.5% were paraphrases. 20% of the reported answers contradicted the finalized answer. Another 30% demonstrated the elaboration relationship with respect to the stabilized answer. Partial overlap and degree of certainty were observed only a few times while attribution did not appear at all.

It should be noted that the distribution of CST relationships is clearly related to the question type. For instance, the majority of questions for which all extracted answers were identical to the final answer related to the time or location of an incident. Two examples are "When did the bombing take place?" (Iraq bombing cluster) and "Where was the plane's destination?" (Egypt Air plane crash story). This is intuitive given that such information is typically known earlier on in a breaking news story, as compared to information related to the cause of an incident or the final number of people killed or injured, which may change or be updated as an investigation unfolds. Unfortunately, for the questions that do not settle on the finalized answers right away, there does not appear to be any common pattern of CST relationships over time.

3.4 Relationships between Vocabulary Usage, Publication Time and Source

In this section, I will describe the analysis I performed in order to examine the relationship between the vocabulary used to express facts in breaking news stories to the respective time and source of a published fact. After describing the hypotheses I tested, I will explain how I created a data set of pairwise answer comparisons from the original set of 135 question and answer sets. Finally, I will discuss the results.

3.4.1 Hypotheses

I tested three hypotheses that concern the relationship between vocabulary usage, and publication time and source.

H1: When comparing a pair of extracted answers to a given question, there is an inverse relationship between vocabulary overlap and publication time difference.

The first hypothesis I tested concerns the relationship between vocabulary usage and publication time difference. I expected to see that, in general, when answers are lexically similar to one another, the publication time difference between them (i.e. between their respective documents) is likely to be smaller as compared to answers that are lexically very dissimilar. This is because over longer periods of time, the fact of interest is likely to have changed, resulting in the usage of new words. Figure 3.3 gives an example from the Milan plane crash cluster. It can be seen that the answers published within smaller time frames of one another are lexically more similar than those that have a large time difference between them.

H2: Answers to a given question that are extracted from different articles published by the same news source, have more shared vocabulary as compared to answers published by different sources.

```
12:22 CNN: no word yet on casualties
12:42 MSNBC: no immediate report on casualties
14:29 MSNBC: at least three people killed
14:52 USA Today: killing at least three people
18:40 ABC News: leaving four dead
```

Figure 3.3: Examples of changing vocabulary over time for the question "How many victims were there?" in the Milan plane crash story.

The second hypothesis concerns the relationship between the lexical similarity of extracted answers and whether or not they were published by the same news source. One reason why answers published by the same source might be more likely to be similar to one another (as compared to those from different sources) is that journalists often use a system of rewrites when covering a breaking story. In other words, they may simply update versions of previously published stories, adding only new information that has become available [81]. To contrast, given the widespread use of text from newswire services [30], we may end up finding that there is not enough variation in vocabulary choice in order to distinguish between the answers published by different sources.

H3: Vocabulary overlap is higher in a set of extracted answers that are paraphrases of one another, versus a set in which there are mutually exclusive answers.

Finally, the third hypothesis considers the difference in vocabulary usage between sets of answers (to a given question) that express the same meaning versus those that contain mutually exclusive answers. While a set of answers with the same meaning may contain many paraphrases, I wished to test the hypothesis that on average, they exhibit a higher degree of lexical similarity than do a set containing mutually exclusive answers. By a set containing "mutually exclusive" answers, I mean a set of answers that could not be considered to report the same answer to the respective question. Figure 3.4 gives two examples of answer sets from the corpus

```
Iraq suicide bombing:
Q: What was the reason for the attack?
A1: to stop the party from participating
in the January election
A2: to intimidate the voters
A3: to threaten the voters
A4: to try to stop the election from
happening

Milan plane crash:
Q: Was it an accident?
A1: Marcello Pera said it "very probably"
appeared to be a terrorist attack.
A2: There were conflicting reports as to
whether it was a terrorist attack or an
accident.
A3: The crash appeared to be an accident.
A4: Authorities said it was an
apparent accident.
```

Figure 3.4: Examples of mutually exclusive answer sets.

that contain mutually exclusive answers. In the case of the Iraq suicide bombing story, the answers express different possible reasons for the attack. Similarly, in the Milan crash example, the answers contradict one another as to whether or not the crash was related to terrorism.

To contrast, Figure 3.5 shows some examples of answer sets that do not contain mutually exclusive answers. In the Iraq suicide bombing example, the answers refer to the same place in different ways. Similarly, in the first Egypt Air example, the answers refer to the same entity (the plane) differently. In the final Egypt Air example, the answers to the question are not mutually exclusive since one answers the question with an absolute temporal expression ("on Sunday") and the other does so with a related temporal expression ("20 minutes after..."). (As illustrated, the third example is one in which my hypothesis does not hold.)

```
Iraq suicide bombing:
Q: Where did the attack take place?
A1: At the gate to the home of the leader
of Iraq's biggest political party.
A2: At the gate of Abdel-Aziz al-Hakim's
compound.
A3: At the gate at the home of Abdul
Aziz al-Hakim.

Egypt Air crash:
Q1: What kind of plane is the Boeing 767?
A1: Boeing 767-300ER
A2: a twin-engine jet
A3: a twin-engine, widebody passenger jet

Q2: When did the search mission begin?
A1: Sunday
A2: 20 minutes after the plane disappeared
from the radar screen
```

Figure 3.5: Three examples of answer sets that are not mutually exclusive.

3.4.2 Data sets

In order to test the three hypotheses, I created two data sets using the corpus of extracted answers. I wanted to make comparisons for all pairs of answers in a given answer set (i.e. the set of answers extracted for a given question). The first data set contains attributes for each of the 42,294 answer pairs over all 135 question/answer sets, that were compared. The second data set contains attributes of the 135 questions in the corpus and their respective answer sets.

First, for each question in the corpus, I compared the extracted answers pairwise with respect to four similarity metrics:

- **Simple cosine:** The cosine similarity using a binary count (1 if a word is shared between two answers, regardless of how many times, and 0 if not).

- **Cosine:** Cosine similarity using idf weights as well as the actual count of tokens in each extracted answer.

- **Token Overlap:** Proportion of shared tokens in both answers.

- **Norm. LCS:** Longest common substring normalized for answer length.

In addition, I found the publication time difference (in minutes) between the answer pair, as well as whether or not they were published by the same news agency.

As potential control variables, I also included the expected answer type, as predicted by a manually created rule-based classifier used in the NSIR question answering system [97]. The expected answer types that appeared in the data were the following: location, number, person, duration, reason, organization, biography, date distance, definition, place and other (those that did not fall into one of the previous categories).

The second data set used in the analysis consists of attributes of each of the 135 questions in the corpus: the expected answer type, the total number of answers found by the judges for that question, the average pairwise similarity (for the five metrics mentioned above), the average publication time difference between answers in the set and whether or not the answer set contained mutually exclusive answers.

3.4.3 Analyses

Here I report how I tested each of the three hypotheses of interest and the results of these tests.

Hypothesis 1: lexical similarity and publication time difference

To test this hypothesis, I used the data set consisting of all pairwise comparisons of answers to questions in the corpus to fit a linear regression model with time difference as the response variable. The independent variables were the four similarity measures (simple cosine, cosine, token overlap and normalized LCS). In addition, I treated the following as control variables: the document cluster to which the answer pair concerned, the expected answer type, and whether or not the two answers were

Variable	**Corr. with TD**
Cluster	-0.038
Answer type	-0.019
Same/diff source	0.021
Sim. cosine	0.038
Cosine	0.028
Token overlap	0.038
Norm. LCS	0.041

Table 3.4: Correlations between independent/control variables and publication time difference.

Indep. var.	**P-value**	**Model R-square**
Sim. cosine	0	0.0032
Cosine	0.00002	0.0032
Token overlap	0	0.0036
Norm. LCS	0	0.0037

Table 3.5: Regression of time difference on each similarity metric with cluster, source and answer type controlled.

published by the same news source.

I first examined the correlations between the independent and control variables and the response variable, publication time difference. The correlation coefficients are shown in Table 3.4. Contrary to my expectations, all four of the similarity metrics have a slightly positive relationship with time difference.

Next, I verified that the time differences between reported answer pairs roughly follows a normal distribution. In order to examine the relationships between the similarity measures and time difference when the effects of source, cluster and answer type are controlled, I fit a linear regression model with each of the four metrics individually as the independent variable, along with the controls. I found that while all of the similarity metrics had a significant linear relationship to time difference (i.e. the coefficients on these variables were significantly greater than 0), none of the models accounted for much of the variance in the response variable.

I also experimented with combining the independent variables and interactions between them or between them and the control variables. However, I did not find any model with an R-squared greater than 0.050. In other words, none of the models explained a significant amount of variance in the dependent variable of publication time difference, therefore, the models would have little accuracy in predicting the time difference between a given pair of extracted answers, given the other variables. However, one interesting observation from the analysis is that the interaction terms between the source control variable (where 1 means the answers came from the same source and 0 indicates they came from difference sources) and all of the similarity metrics was always positive and significant .

I concluded that overall, there is a slight positive relationship between lexical similarity (similar vocabulary usage) and time difference between answers to a given question, so I reject my original hypothesis. However, the source of the answers is an important confounding variable as is the expected answer type. In addition, I concluded that it is unlikely that we will be able to build a model to predict the publication time difference for a given pair of answers to a question, based only on their lexical similarity, publishing source and expected answer type.

Hypothesis 2: lexical similarity and news source

To test whether or not extracted answers published by the same news source are generally more lexically similar as compared to answer pairs from different sources, I conducted a t-test for each of the similarity metrics. The mean similarity between answers for each group (same source answers vs. those from different sources) and the p-value for the one-sided hypothesis test are shown in Table 3.6. My conclusion with respect to the second hypothesis is that answer pairs published by the same source have more shared vocabulary than do answer pairs published by different news sources. This is true for all four of the metrics I tested.

Similarity measure	Mean - same source	Mean - different sources	P-value
Simp. cosine	0.392	0.312	0
Cosine	0.392	0.312	0
Token overlap	0.327	0.232	0
Norm. LCS	0.355	0.264	0

Table 3.6: T-tests for the comparison of mean similarity between answer pairs published by the same news source vs. those published by different sources.

Attribute	Mean - not mut. exc.	Mean - mut. exc.	P-value
Answers found	13.8	22.8	0.005
Simp. cosine	0.578	0.334	0
Cosine	0.573	0.310	0
Token overlap	0.509	0.258	0
Norm. LCS	0.552	0.291	0

Table 3.7: T-tests for the comparison of mean similarity between answer pairs for questions in which there are not mutually exclusive answers vs. sets in which some answers are mutually exclusive.

Hypothesis 3: lexical similarity and mutual exclusivity of answer sets

To test the third hypothesis, I used the data set consisting of attributes of the 135 questions in the corpus. I divided the questions up into those that did not contain mutually exclusive answers and those that did. My hypothesis was that answer sets containing mutually exclusive answers, on average, should exhibit less vocabulary overlap as compared to answer sets in which the same meaning is expressed. The average answer pair similarity, as well as the number of answers found per question, and the p-value for the t-test comparing the means between groups are shown in Table 3.7.

Clearly, on average, answers for a given question that express similar information (are not mutually exclusive) exhibit more lexical similarity as compared to answers from sets where some answers are mutually exclusive. In addition, the number of answers found for a question was typically greater in the sets containing mutually exclusive answers, as compared to the sets of answers expressing the same meaning.

The analysis suggests that there is no direct relationship between lexical similarity and publication time difference between a given pair of answers to a question, independent of other factors such as the source and the type of question. This is logical given that journalists often repeat information that has already been reported and the widespread use of newswire sources. There is, however, evidence of clearer relationships between lexical similarity and source. On average in the corpus, answer pairs for a given question that are published by the same source are more similar than those coming from different sources. In addition, there was a clearly more similarity between answer pairs that expressed the same meaning (were not mutually exclusive) as compared to those in which different meanings were expressed as an answer to the same question.

3.5 Discussion

Analyzing the extracted answers from the corpus has illustrated some challenges for tracking facts in online, breaking news. For example, while almost 75% of the reported answers in the corpus were correct at the time that they were reported, less than 15% represented the final, stabilized answers to the respective question. This means that on average, in the traditional question answering scenario where one "best" answer is returned to the user, 85% of the potential answers to be found by a system are not the finalized (most accurate) answer. Likewise, as shown in the semantic analysis of the answers, finding the most appropriate answer is challenged by the fact that there are subtle yet non-trivial relationships between answers, such as paraphrase and elaboration. In fact, it will be shown in Chapter V that when given a set of sentences that are relevant (contain answers for) a given question, human annotators do not agree on which are new (previously unreported) answers.

It was previously mentioned that one possible design for an IR system for the task of interest, short-term event tracking, would be to present all extracted answers to a question to the user. Ideally, if the system is capable of finding all reported answers in the news articles to the user, we can expect about 25% of the answers to convey incorrect information. Showing all possible answers also has the advantage of allowing the user to see how the information reported by sources that he or she trusts the most compare to that reported by other agencies that may have a different reporting bias. In addition, seeing what all sources report might give the user an idea of how accurate the reported information is.

In the third analysis, it was not surprising to find that on average, answers reported by the same source have more vocabulary in common than do answers from different sources. This is in agreement with previous research on the relationship between source and textual similarity (e.g. [70, 30]). It was also found that the similarity of two answers is not closely related to their publication time difference, even when other factors such as source and question type are controlled. Because of the widespread use of newswire to cover breaking news stories, I expected to see that overall, answers to factual questions reported within a small timespan of one another would have a large vocabulary overlap, while answers reported far apart from one another in time would tend to be less lexically similar. However, it appears to be the case that the degree of lexical similarity, as measured by the four simple measures I used, does not change dramatically in a single fact over time. In addition, it may also be the case that in a breaking news story, where journalists need to deliver information as quickly as possible in order to compete with other news agencies, they may not refresh the portions of the story that do not change from one point in time to another. Therefore, we may observe changes in vocabulary only at the points in time where the facts have changed, rather than uniformly over time.

CHAPTER IV

Recovering Chronological Relationships in Dynamic Information

As previously discussed in Chapter I, when an important event happens, large numbers of news sources report on it. In doing so, they draw information from direct participants in the event, eyewitnesses, official reports, copy from the newswire, as well as from each other. As anyone who follows an event can attest, often multiple sources present complementary accounts of the news. Each source has its own reputation, biases, and agenda. In addition to source, news accounts of an event vary over time. Often initial reports turn out to be partially or fully incorrect. It takes a certain amount of time for accounts to stabilize and to be accepted as the ground truth.

In considering how information evolves over time and is expressed through text, I have examined sets of documents on the same story published over time by multiple news agencies, and have found that they exhibit a number of interesting relationships. For example, a given pair of related documents may express some of the same factual information and yet each may contain novel information that the other does not. An example with respect to a single fact is illustrated in Figure 4.1. The sentences shown were extracted from documents about the Milan place crash story, which describes the crash of a small plane into a skyscraper. The sentences all concern the location from where the plane departed. They are shown with their respective publication times and sources in chronological order.

```
04/18/02 13:17 (CNN)
The plane, en route from Locarno in Switzerland,
to Rome, Italy, smashed into the Pirelli building's
26th floor at 5:50 p.m. (1450 GMT) on Thursday.

04/18/02 13:42 (ABCNews)
The plane was destined for Italy's capital Rome,
but there were conflicting reports as to whether it
had come from Locarno, Switzerland or Sofia, Bulgaria.

04/18/02 13:42 (CNN)
The plane, en route from Locarno in Switzerland,
to Rome, Italy, smashed into the Pirelli building's
26th floor at 5:50 p.m. (1450 GMT) on Thursday.

04/18/02 13:42 (FoxNews)
The plane had taken off from Locarno, Switzerland,
and was heading to Milan's Linate airport,
De Simone said.
```

Figure 4.1: Dynamic information example.

In short, following information in a news story over time and across sources is a challenging task due to the dynamic nature of such texts. As facts, beliefs and opinions surrounding an event change, so do the texts that report on them. In other words, such stories can be viewed as "evolving" over time, beginning with the information reported in the first story that makes the news. Currently, I will attempt to model these phenomena using a phylogenetic approach. In phylogenetics, the history of a set of species is reconstructed, under the assumption that they evolved from a common ancestor, with genetic mutations occurring at different points in time. The "species" I will study are related documents describing the same news story.

In addition, I will test a second approach that is inspired by language modeling. I will use a language model generated from the earliest document in each set, to chronologically order the remaining documents. In doing so, I hypothesize that as time goes on and the story changes, the likelihood that the original language model could have generated a later document should decrease. In both experiments, I evaluate the fit of the evolutionary models with respect to their ability to recover the chronological relationships between the documents in a given cluster. Rather than experimenting with a large number of text representation methods within each approach, I have applied the same preprocessing techniques to the texts in the corpus before implementing the models. It is likely that I will be able to improve the performance of both approaches on the chronology recovery task in future work. However, the goal of the current paper is to evaluate the extent to which multi-document clusters of news articles exhibit evolutionary properties as well as to see which approach, phylogeny or language modeling, is more promising for modeling inter-document dynamics.

4.1 Related Work

Before describing the experiments I conducted I will review some previous research that was not discussed previously in Chapter II. The work that is briefly discussed here is directly related to phylogenetic analysis. I will also note how my current approach differs from those taken in previous studies.

4.1.1 A method for phylogenetic analysis

The Fitch-Margoliash method is used in the biological sciences for constructing a phylogenetic tree for a set of species, based on sequences of amino acids found in their DNA [43]. First, mutation distances are calculated between each pair of species. This distance is the minimum number of sites that would have to be changed in order for one string to mutate into the other. Initially, each of the N species is assigned to its own subset, such that there are N subsets. They are then joined together, starting with those that have the smallest mutation distance between them, such that the number of subsets is reduced by one at each cycle, until all subsets have been joined to the tree.

Because of the manner in which the initial sets are chosen, various phylogenetic trees will result from the different initial assignments. Therefore, it is necessary to test between alternative trees. For each tree, one sums over the distances between each pair of species, resulting in a new distance matrix that can be compared to the original mutation distances. The "percent deviation" of the reconstructed values in the tree from the original input distances are found by summing the squared percent change for each species. For example, if the original mutation distances between pairs of species are in the upper triangle of the distance matrix, while the new distances according to the candidate tree are in the lower triangle, then for each species pair the original distance is (i, j) and the new distance is (j, i).

$$\text{Percent deviation} = \sum_{i<j} (\frac{|(i,j) - (j,i)|^2}{(i,j)}) * 100$$

Seeking the statistically optimal phylogenetic tree from the set of all possible trees involves minimizing the percent deviation.

4.1.2 Phylogenetic trees and text analysis

Bennett and colleagues applied phylogenetic inference algorithms to reconstruct the evolutionary history of 33 chain letters collected between 1980 and 1995 [17]. Because the chain letters circulated before the widespread use of email, they proposed that the letters mutated and evolved as generations of receivers photocopied them until no longer legible. At such a point, the next recipient would likely retype the letter, introducing new errors and variations.

The distance metric between each pair of chain letters x and y used in constructing the tree was the amount of information, $d(x, y)$ shared by the pair of letters. Once the distance matrix was computed, the authors used various methods, including Fitch-Margoliash, in constructing phylogenetic trees. The tree was rooted using the letter with the earliest known date. Using the same distance metric, the various methods for constructing the tree yielded similar trees. Once the tree was constructed, the authors were able to explain how the chain letters evolved over time. For example, names of individuals and the dates of different events mentioned in the letter (such as the death of someone who broke the chain) changed at different points in its evolution. In addition, new "genes" often appeared. The resulting tree was almost a perfect phylogeny, as the authors were able to confirm that letters containing the same characteristics were always grouped together.

4.1.3 Current approach

The current work is inspired by Bennett's research but differs in some important ways. In the chain letters, mutations occurred over time because of letters being recopied by recipients, who might misspell or misinterpret words in the letter when preparing copies to mail out to the next receivers. Alternatively, details of the letters were occasionally changed deliberately. For example, when the letters were first brought to the U.S. from Europe, certain names and titles were changed. In the current work, I assume that over time, I will observe mutations in news stories because they reflect events and facts in the real world that are constantly changing.

There are some other interesting nuances in the current problem. For example, while I assume that the texts I observe express the facts in the world, there is rarely only one way to express the same concept or fact in natural language. Therefore, I expect to encounter many instances of paraphrases in the data. At the same time, it is known that journalists use newswire sources and may also copy large parts of previously published news stories in creating an update on a given situation [30, 81]. Therefore, I will also observe many instances of identical expressions, published by different sources and perhaps even at different points in time.

In the experiments, I attempt to recover the chronological relationships between related documents using two different approaches. In the first approach, I create an unrooted phylogenetic tree for each document cluster, and then reroot each tree at the document in the cluster that has the earliest publication date. Therefore, S1 (Species 1) is at the base of the tree, and I propose that the remaining documents arise as mutations occur. Once I have the rerooted tree for a cluster of documents, I calculate the distance from the root, S1, to each of the other documents. The hypothesis is that these distances should correlate well to the chronological ordering of the documents.

I will compare the performance of the phylogenetic document ordering algorithm to that of a second approach based on language modeling. Language modeling has been used extensively in information retrieval for document ranking. In this setting, a document is considered to be relevant to an information query if the language model built from the document assigns a high probability to the query [93]. More recently, [64] used language models for modeling inter-document relationships. In the experiments, I create a language model from the earliest document in each cluster. I then evaluate it on the remaining documents and use its fit to rank them. Our hypothesis is that the model fit should be better for the earlier documents and degrade as time goes on, since as the facts in the story change, new terms and expressions arise.

4.2 Corpus

Table 4.1 shows the characteristics of the document clusters used in the experiments. Six clusters were collected manually, three clusters (Bali bombing, Turkish Air crash and Hamas bombing) were collected automatically from a Web-based news tracking system and 27 clusters were taken from the TREC Novelty Track 2003 and 2004 test sets [116] [1]. They were randomly assigned to the training (15 clusters), development/test (6 clusters) and test data sets (15 clusters), although I did ensure that they were distributed to each data set rather evenly by type (manually collected, automatically collected and TREC clusters).

As can been seen, the Novelty clusters differ from the manually collected clusters in one important way. While the manual clusters were collected over a relatively short time period (e.g. a few days), the Novelty clusters typically contain documents published over a much wider time span. In addition, the manually collected clusters all describe emergency news stories (e.g. plane crashes, fires), while the Novelty clusters include a wide range of topics. For use in the experiments, all texts in the corpus were tokenized, such that all punctuation was removed and all capital letters were made lowercase.

4.3 Phylogenetics Experiments

In this section, I will discuss how the phylogenetics experiments were conducted. In particular, I will explain how the phylogenetic trees were used to order a given set of documents. In addition, I will provide an example to illustrate this process.

[1] I included Novelty clusters that were labeled as describing events only. Opinions clusters were excluded

Story	**Doc.**	**Time span**	**Sources**	**Data set**
Milan plane crash	56	1.5 days	5	train
RI nightclub fire	43	1.5 days	8	train
Iraq bombing	30	1.5 days	10	train
Turkish Air crash	10	6 days	4	train
N4 - EgyptAir crash	25	8 months	3	train
N6 - Unabomber	25	3.5 years	3	train
N8 - Berenson imprisoned treason	25	4.5 years	3	train
N33 - Russian submarine sinks	25	1 month	3	train
N34 - Shuttle Discovery	25	1 month	3	train
N42 - JFK Jr. dies	25	1 year	3	train
N43 - Chinese earthquake	25	1 year	2	train
N44 - Plane gondola accident	25	1 year	2	train
N51 - Pinochet arrested	25	10 months	3	train
N64 - Japan nuclear accident	25	1 year	3	train
N87 - Birmingham church bomb	27	4 years	3	train
Columbia shuttle disaster	41	2.5 days	6	devtest
Bali bombing	10	13 days	5	devtest
N7 - Atlanta Olympics bombing	25	3.5 years	2	devtest
N49 - 1998 Nobel peace prize	25	3 months	2	devtest
N53 - Death of James Byrd, Jr.	32	1.5 years	2	devtest
N81 - Matthew Shepard	25	1.5 years	2	devtest
GulfAir plane crash	11	1 month	7	test
Honduras bus hijacking	46	2 days	10	test
Hamas bombing	11	2 days	7	test
N9 - Columbine shooting	25	1 year	3	test
N11 - Hurricane Mitch	25	2 months	2	test
N16 - Kenya embassy bomb	25	1 year	3	test
N37 - Olympic bribe scandal	25	2 years	3	test
N40 - Wen Ho Lee, Los Alamos	25	1 year	3	test
N45 - Slepian abortion murder	25	1.5 years	2	test
N48 - Human genome decoded	25	2 years	3	test
N50 - Balloonist solo flight	25	1 year	2	test
N59 - Steward plane crash	25	1 year	3	test
N69 - Concorde crash	27	2 months	3	test
N80 - Turkey earthquake	41	4.5 years	2	test
N83 - Marine Osprey	25	5 months	3	test

Table 4.1: Document clusters used in experiments.

4.3.1 Document ordering

I applied the phylogenetic technique on the full text of the documents, as well as on summaries produced from each individual document using various compression rates using the MEAD extractive summarizer [101]. The intuition behind using summarization is that it might highlight the most salient information in each document, while eliminating some information that might not be important for recovering inter-document relationships. For each run on a given document cluster, I calculated the Levenshtein matrix, or the edit distances between all pairs of documents (at the word level). This was used as the mutation distance in order to construct the phylogenetic trees using the Fitch-Margoliash method. I used the Fitch program (part of the Phylip Inference package) to construct the trees [40].

Since Fitch produces unrooted trees, such that one obtains relative distances between documents, rather than from a common starting point, I rerooted each tree at the earliest sentence in the cluster. The text dynamics rerooting algorithm is shown in Algorithm 1.

4.3.2 An example

In this section, I illustrate the methods using a small example cluster of four topically related documents from the Milan training cluster. For illustrative purposes, I have represented each document as one sentence extracted from it, rather than showing the entire text of the document. Each document species is shown with its respective publication date, time stamp and source in Figure 4.2.

First, the Levenshtein matrix is calculated, yielding the distance matrix for Fitch. The distance matrix for the above example is shown in Figure 4.3. Each entry (i, j) in the matrix shows the word-level edit distance between document i and j. Note

Algorithm 1 TD tree rerooting algorithm.

```
Root tree at $S_1$
$depth(S_1) = 0$
Initialize stack $q$ of next documents to process
Push $S_1$ onto $q$
repeat
  $S_i$= next element in $q$
  $seen(S_i) = 1$
  Find depth of $S_i$ in tree
  depth($S_i$)=Find_depth($S_i$)
until stack $q$ is empty

Function Find_depth($S_i$)
for each element $a_i$ in tree do
  $b_i$ is element adjacent to $a_i$ and $distance(a_i, b_i) = c_i$
  if $a_i$=$S_i$ and seen($b_i$)=0 then
    Push $b_i$ onto $q$
    depth($b_i$)=$c_i$ + depth($S_i$)
    Return depth($b_i$)
  end if
  if $b_i$=$S_i$ and seen($a_i$)=0 then
    Push $a_i$ onto $q$
    depth($a_i$)=$c_i$ + depth($S_i$)
    Return depth($a_i$)
  end if
end for
```

that the Levenshtein matrix is also symmetric with zeros along the diagonal.

Once the best fitting evolutionary tree is found by the Fitch-Margoliash method, it is then rerooted at the earliest document in the cluster. The unrooted tree (output of Fitch) for the example is shown in Figure 4.4. Note that the tree shows both the document species as well as internal nodes, intermediate points at which a mutations occur. The nodes and species are shown with their respective distances from node I_1, an arbitrary point. The corresponding rerooted tree is shown in Figure 4.5. Here, the distances shown are from the given node or species to S1, the root. To obtain these distances, the tree is traversed from the root out. The system ranking is then determined with respect to the distances, with species closer to the root having higher ranks. The ranks correspond to the chronological ordering of the document species. To evaluate, the system rankings are compared to the actual chronological ordering

S1: Italian TV says the crash put a hole in the 25th floor of the Pirelli building, and that smoke is pouring from the opening. (04/18/02 12:22, CNN)

S2: Italian TV showed a hole in the side of the Pirelli building with smoke pouring from the opening. (04/18/02 12:32, CNN)

S3: Italian state television said the crash put a hole in the 25th floor of the Pirelli building. (04/18/02 12:42, MSNBC)

S4: Italian state television said the crash put a hole in the 25th floor of the 30-story building. (04/18/02 12:44, FOX)

Figure 4.2: Sample document "species" in chronological order.

	S1	S2	S3	S4
S1	0	10	12	13
S2	10	0	15	16
S3	12	15	0	1
S4	13	16	1	0

Figure 4.3: Levenshtein matrix for 4 input document species.

of the documents. Figure 4.6 illustrates this process.

4.4 Language Modeling Experiments

As previously mentioned, for each document cluster, a language model was built from the earliest document in the set. More specifically, a simple trigram backoff language model with Good Turing discounting was created and evaluated against every other document in the cluster using the CMU-Cambridge toolkit [28]. Since the first document in a cluster typically had a much smaller vocabulary than latter documents, I used the out-of-vocabulary (OOV) rates as well as the backoff event information rather than model perplexity in order to assess the fit with respect to each document in the cluster. I hypothesized that for documents published later on, the OOV rate should be greater. Likewise, I expect to see more backoff events, such that the trigram-hit ratios should be smaller, and unigram-hit ratios larger, as compared to earlier documents. There were three experiments per cluster: one in which documents were ordered by OOV, by unigram-hit ratio and by trigram-hit

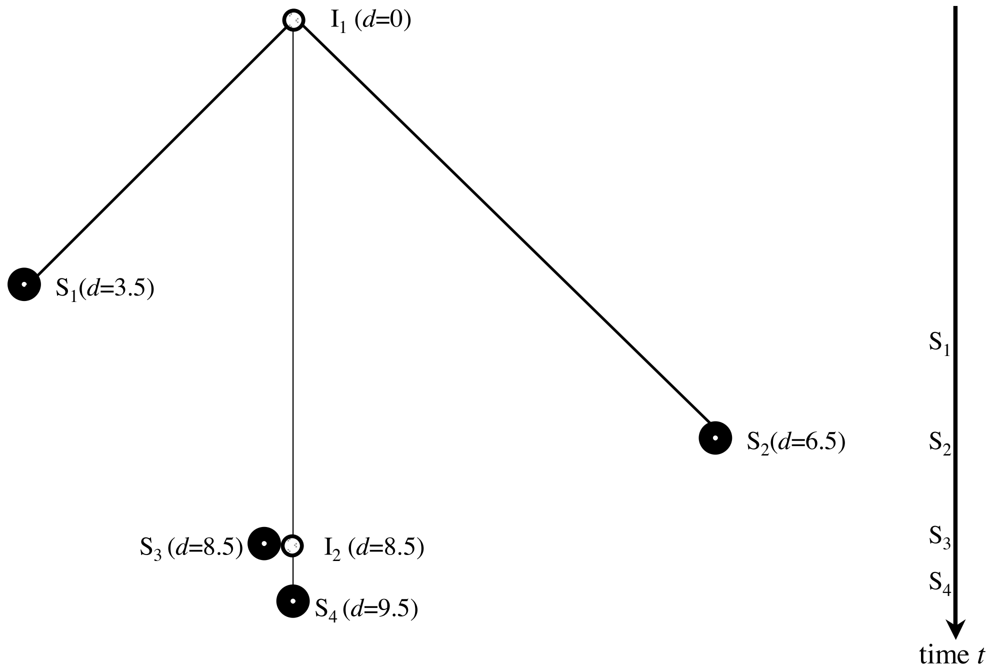


Figure 4.4: Unrooted tree.

ratio (ranked in reverse order). I then compared the system orderings to the true orderings in the same manner as in the phylogenetic experiments.

4.5 Experimental Results

In the current section, I will explain how the different documents orderings of the various systems were evaluated. I will also present the results on each phase of the experiment.

4.5.1 Evaluation method

For each cluster and system ordering, the Kendall rank-order correlation coefficient was calculated [115]. Kendall's τ quantifies the extent to which the rankings assigned by the system are correlated to the actual rankings: $\tau = \frac{2*(n_a - n_d)}{N(N-1)}$, where

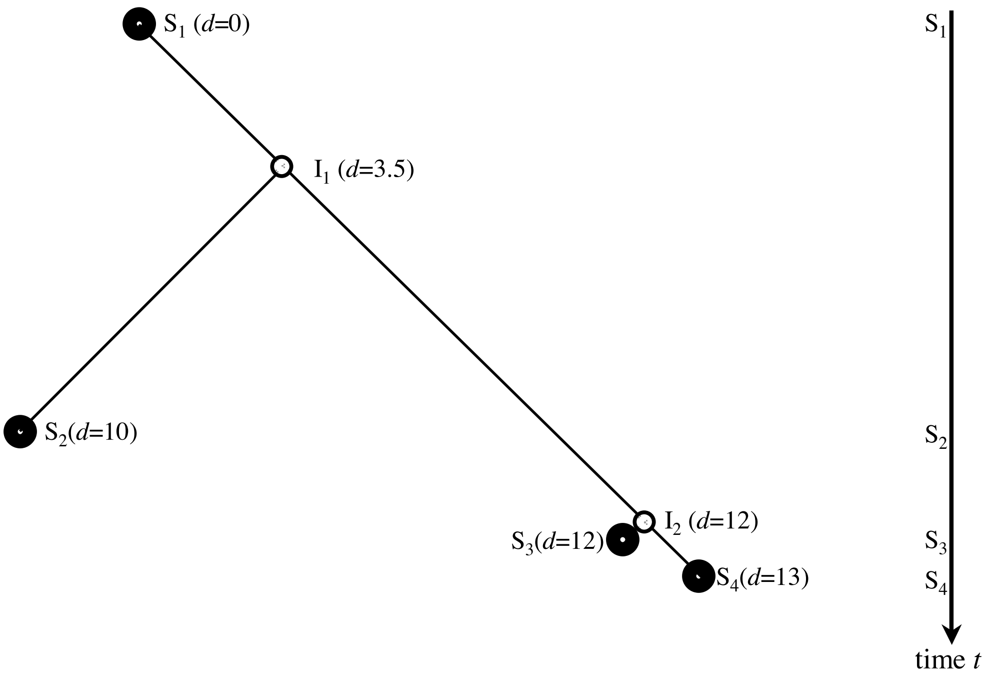


Figure 4.5: Tree rooted at Species 1 (S1).

Document species	**Distance from root**	**System rank**	**Actual rank**
S1	0	1	1
S2	10	2	2
S3	12	3	3
S4	13	4	4

Figure 4.6: Chronological ordering of the input documents.

n_a is the number of agreements, n_d is the number of disagreements and N is the number of ranked documents. In the case of tied ranks, there is an adjusting factor in the denominator, such that that penalty is less for a disagreement between the system and the actual ranks.

Essentially, τ is the ratio of the difference between the number of partial ranks in agreement and those in disagreement between the system and the actual rankings to the maximum possible total. Therefore, a τ of 1 indicates that the ranks assigned by the system agree perfectly with the true ranks. Figure 4.7 illustrates the calculation of τ for the set of example document species.

System	Actual
S1 > S2	S1 > S2
S1 > S3	S1 > S3
S1 > S4	S1 > S4
S2 > S3	S2 > S3
S2 > S4	S2 > S4
S3 > S4	S3 > S4

Figure 4.7: Comparing partial rank orderings for calculating τ.

Comparing the partial rankings of the system to the actual rankings, there are 6 in agreement and none in disagreement. Therefore, $\tau = \frac{2*(6-0)}{4*(4-1)} = 1$.

The p-value for a τ of 1 when N=4 is 0.025. The interpretation of this value is that if we repeatedly draw a sample of four documents from the population of documents related to the Milan story, then under the null hypothesis that the rankings assigned by the algorithm and the actual rankings are uncorrelated, the probability of finding a τ=1 (or a more extreme value of the test statistic) is 0.025. Currently, I will use a significance level of 0.10 for reporting the experimental results.

4.5.2 Training phase

In the training phase, I evaluated 11 document ordering mechanisms on the 15 training clusters. I implemented the phylogenetic algorithm on the full text of the documents, as well as on the document summaries at lengths of 1, 2, 3, 4, 5, 6 and 8 sentences. I also evaluated document ordering using the three language modeling approaches (based on trigram-hit and unigram-hit in the backoff model, and OOV as previously discussed). The median Kendall's τ over the 15 document clusters, and the number of clusters on which τ was statistically significant are shown in Table 4.2. Over all clusters, the language modeling OOV approach performed the best, having a median τ of 0.28. In addition, for 13 of 15 training clusters, the results were

	Med. τ	**# Sig.**
Full doc	0.16	8/15
Summ-1	0.13	6
Summ-2	0.12	5
Summ-3	0.13	6
Summ-4	0.16	6
Summ-5	0.17	6
Summ-6	0.09	6
Summ-8	0.12	6
3gram	0.17	7
1gram	0.21	11
OOV	0.28	13

Table 4.2: Median τ and the number of data clusters with a significant result.

	Med. τ	**# Sig.**
Summ-5	0.05	3/11
1gram	0.20	8/11
OOV	0.19	8/11

Table 4.3: Median τ and the number of clusters with a significant result for the 11 Novelty training clusters.

statistically significant.

The best run for the phylogenetic approach was the one which calculated the edit distance between each document species based on the 5-sentence summary of each document. Tables 4.3 and 4.4 show the comparison of this approach against the two best language modeling approaches (1gram and OOV) on the 11 Novelty data clusters and the 3 manually-created clusters, respectively. As mentioned in Section 4.2, the manual clusters differ from the Novelty clusters not only in that all discuss emergency news topics (e.g. that are likely to report changes rapidly over time) but also in that the publication times of the documents are relatively closer together. Here we can see that on the manual clusters, all three methods yield statistically significant results on all three manual clusters. However, for the Novelty clusters, 1gram and OOV perform much better than the phylogenetic technique.

	Med. τ	**# Sig.**
Summ-5	0.32	3/3
1gram	0.42	3/3
OOV	0.26	3/3

Table 4.4: Median τ and the number of clusters with a significant result for the 3 manual training clusters.

Cluster	**OOV**	**1gram**	**Summ-4**	**Summ-5**
Columbia shuttle	0.56	0.52	0.46	0.48
Bali bombing	0.20	0.24	0.51	0.29
N7 - Olympics bombing	0.32	0.27	0.15	0.24
N49 - Nobel prize	0	0.29	0.25	0.31
N53 - Death of J. Byrd	0.21	0.27	0.04	0.20
N81 - Matthew Shepard	0.35	0.23	0.04	0.19
Med. τ	0.26	0.27	0.20	0.26
# Sig.	4/6	5/6	3/6	5/6

Table 4.5: Individual cluster τ, and median τ and significance for all 6 dev/test clusters.

4.5.3 Development/test phase

In the development/test phase, I evaluated the top two language modeling approaches (1gram and OOV) as well as the best two phylogenetic techniques (Summ-4 and Summ-5) in order to distinguish them further in terms of performance. Table 4.5 shows the τ for each of the six development/test clusters as well as the median over all clusters and the number of significant orderings. In this set, only one cluster, which describes the Columbia shuttle disaster, is a manually-created cluster and as expected, all four techniques achieve a statistically significant result on ordering the 41 documents in the cluster. However, I again observe some poor performances on the Novelty clusters. In particular, Summ-4 achieves a τ of only 0.04 on clusters N53 and N81. Given its lower median τ as well as having a significant performance on only half of the clusters, we eliminate Summ-4 and evaluate the remaining three techniques on the unseen test data set.

	Med. τ	**# Sig.**
Summ-5	0.15	5/15
1gram	0.14	6/15
OOV	0.22	9/15

Table 4.6: Median τ and the number of clusters with a significant result for 15 test clusters.

4.5.4 Test phase

The performance of the three remaining techniques is shown in Table 6.7. The technique that orders documents with respect to their OOV rate when evaluated against the language model created by the earliest document in the set outperformed the other two methods. In particular, the OOV technique achieved a statistically significant Kendall's τ on 9 of the 15 unseen test clusters.

4.6 Conclusions

While over all data clusters, the OOV technique outperformed all others, we have also seen that in general, better results were obtained on the manually-collected document sets as compared to the Novelty clusters. Table 4.7 shows the performance of the OOV (language model) and Summ-5 (phylogenetic) techniques the six manual clusters over all data sets. To contrast, over all 27 Novelty clusters in the corpus, the median τ for the OOV and Summ-5 techniques was 0.22 and 0.17, respectively. Therefore, one conclusion from the experiments is that the evolutionary models that I have proposed and implemented fit the manual clusters rather well. As previously mentioned, these clusters were collected over shorter periods of time from Web-based news sources. In addition, I tried to collect as many documents as possible that were published over time describing the given subject, which was an emergency situation.

To contrast, the Novelty cluster topics are more varied and as can be seen in Table 4.1, the publication time spans are typically larger (e.g. over months or years) rather than over days, as in the manual clusters. It is obvious that the evolutionary

Cluster	**OOV**	**Summ-5**
Gulfair plane crash	0.37	0.39
Honduras bus hijacking	0.12	0.17
Columbia shuttle	0.56	0.48
Milan plane crash	0.26	0.33
RI nightclub fire	0.58	0.32
Iraq bombing	0.24	0.17
Med. τ	0.31	0.33
# Sig.	5/6	6/6

Table 4.7: Performance over all 6 manually-created clusters.

models in general, do not fit these types of document clusters as well. In fact, the poorest performances observed in the test data are on Novelty clusters. For example, for the cluster N80 about the Turkey earthquake, which contains 41 documents published over a period of 4.1 years, none of the techniques achieves a statistically significant result. Therefore, I conclude that the evolutionary models fit well and are most useful for predicting relationships between documents describing related, breaking news stories and that are published over shorter time intervals.

CHAPTER V

Fact and Topic-focused Judgments of Relevance and Novelty: an Annotation Experiment

A good deal of research in information retrieval has concerned the problem of identifying relevant and novel information in topically related documents published over time. The automatic detection of the textual units that contain new information, or information that the user has not yet previously encountered, would be of benefit to a number of IR applications. While finding information that is relevant to a user's information need, IR systems incorporating "novelty detection," also aim to reduce the amount of redundant information seen by the user.

While often not explicitly referred to as novelty detection, much previous work considers, in general, the problem of distinguishing new information from that already seen. Many researchers have addressed this problem at the document level. For example, the First Story Detection (FSD) task in the Topic Detection and Tracking (TDT) initiative [3] is an example of detecting novel information at the document level. The goal of FSD is to identify, in a stream of broadcast news stories, those that introduce a new story that has not been discussed previously. Similarly, another TDT task, Link Detection [42], can be viewed as a binary novelty problem, as it involves deciding whether or not an input pair of documents discusses the same news story. The concept of novelty at the document level has also been considered in the context of information filtering (e.g. [129]), where the objective is to find documents that not only match a user's information profile but that also contain previously unseen information.

In addition, there has been an interest in novelty detection at smaller levels of textual granularity and in particular, at the sentence level. For instance, Allan and colleagues proposed δ-summarization, in which summaries of an incoming stream of documents are produced over time, highlighting what has changed since the last summary was produced [5]. Their goal was to find "interesting" sentences that were both useful (relevant to a desired topic) and novel (contained information not present in previously seen documents).

It is clear that a means for detecting relevant yet novel information at the sentence level would be of direct benefit to the many IR systems that operate at the sentence level, such as extractive text summarizers. In particular, controlling the amount of redundancy while still choosing the most relevant sentences is a well-known problem in summarization [47]. In addition, novelty detection would be useful in the context of question answering systems that, after having identified documents relevant to the input question, then find relevant sentences and perform answer extraction from the selected sentences (e.g. the NSIR system [98]). For example, such systems could use novelty detection to determine which sentences contain the same answer to a given question.

5.0.1 Sentence-level novelty detection

Two major research initiatives have specifically focused on the sentence-level novelty detection problem, and both have noted several problems that have hindered further progress on this task. A 1999 summer workshop on "Topic-based Novelty Detection" had the goal of addressing both the First Story Detection as well as the

"New Information Detection" problem [6]. While in FSD, the aim is to identify the onset of a new story in a stream of news, in the new information detection task the idea is to prevent users from becoming overwhelmed by repetitive messages. In comparison to FSD, this task operates at the sentence level, and within a given story rather than across stories. In the workshop participants' final report [6], they noted that not much progress was made towards the latter of their two goals. The main problem they cited is that the meaning of "novel information" is very difficult to define precisely and is quite context-dependent.

A second major effort towards sentence-level novelty detection was the TREC Novelty Track, which was held in 2002, 2003 and 2004 TRECs[1]. In this evaluation, the goal was to train systems that perform a two-stage task. Given a TREC topic query and a set of documents relevant to the topic, the systems should first retrieve all sentences that are relevant to the stated topic. In the second step, the systems were to choose, from the list of relevant sentences, the novel sentences, defined as those containing "previously unseen information" [116]. Several problems were noted by the organizers in creating the manually-labeled data sets for the training and evaluation of the systems. In the annotation process, the assessors were presented with 25 documents relevant to their given topic, and then asked to carry out the two-stage sentence-level relevance and novelty detection as described above. One problem noted in the 2002 evaluation was that the assessors chose very few relevant sentences, which resulted in many negative and few positive relevance examples available for training the systems [49]. At the same time, most of the relevant sentences were also marked as being novel. Finally, a "major assessor effect" was noted. In short, the annotators typically did not choose the same proportion of relevant and novel

[1] $http://trec.nist.gov$

sentences from a given set of documents, nor did they tend to choose the same sentences.

In the 2003 evaluation, several changes were made to the manual annotation process [116]. Most notably, for each topic, one assessor (deemed the official assessor) created the set of the 25 most relevant articles, by searching a document collection. The documents were then ordered chronologically, and the assessor performed the two-stage manual retrieval of relevant and novel sentences. A second assessor also performed the sentence-level annotation task in order to assess interjudge agreement. This time, the distributions of sentences marked as relevant and novel were more reasonable (more sentences were marked as relevant, and fewer relevant sentences were marked as novel). However, a large assessor effect was again noted. While the interjudge agreement was not reported, it was noted that the judges in general did tend to pick approximately the same numbers of relevant and novel sentences for a given cluster, but they did not tend to pick the same sentences.

5.0.2 Variations in relevance judgments

Researchers in information retrieval have long noted the challenges associated with using relevance judgments. While the concept of relevance is essential for the development and evaluation of IR systems, its nature is still not well understood, nor is it always clear how to operationalize relevance within a given system [44, 82]. It is also well known that human judgments of relevance vary, both across multiple judges and over time by the same judge [110], leading some to criticize the use of such judgments (e.g. [34, 51]). In terms of using relevance judgments to evaluate IR systems, several studies have suggested that the variation across assessors does not significantly alter the resulting system rankings. For example, in an experiment using the TREC-4 data set, Voorhees found that the resulting system rankings produced using relevance judgments collected from different assessors were highly correlated [126]. Therefore, from a systems evaluation perspective, how well assessors agree on relevance judgments may not be of a great concern.

However, from a system building standpoint, low interjudge agreement is more of a problem. This is because in order to be able to build systems that replicate human judgment on a certain task, one must first verify if humans themselves produce similar judgments [24]. In fact, in evaluating machine learning classification approaches, agreement between independent judges on annotation tasks typically represents an upper bound for the performance of systems that assign labels or classes automatically (e.g. [78, 122]). This suggests that in order to make progress in developing systems for sentence-level relevance and novelty detection, it is desirable to start with manually annotated data on which a satisfactory level of interjudge agreement has been established.

5.0.3 Fact-focused relevance and novelty detection

It has previously been stated that further progress in novelty detection has not been made because what novelty means is too undefined [6]. In addition, it has been noted that sentence-level relevance detection is a very difficult problem, and that novelty detection performance is, of course, directly dependent on its performance [7]. Therefore, the goal is to try to address these problems.

Currently, I propose a new sentence-level relevance and novelty annotation task, and will evaluate its reproducibility in an experiment. The task I propose is that of fact-based relevance and novelty detection. I assume that a user has a general topic of interest, and has identified a set of documents (ordered chronologically) relevant to that topic. Next, I assume that the user has a set of facts of specific interest about the topic. For simplicity, the user may state each fact of interest as a natural language question. The task is, for a given fact, to first identify the set of sentences in the document set that contain relevant information. A sentence contains relevant information if it provides an answer to the factual question. (However, the answer need not be a correct or definitive answer.) In a second step, only the sentences containing unique (previously unseen) information about the fact of interest are kept.

I am interested in evaluating the new task for several reasons. First, I hypothesize that I will obtain satisfactory levels of interjudge agreement on the fact-based relevance annotation task. Previous studies have evaluated the agreement between annotators for identifying fact-like semantic units in text with some promising results. For example, both van Halteren and Teufel [124] and Nenkova and Passonneau [84] focused on developing measures to be used in evaluating the content of automatically produced text summaries. In the first study, independent annotators identified the factoids contained in each of 50 texts (summaries of a single news article). Factoids were defined as “atomic information units” that are represented in First Order Predicate Logic style semantic expressions. The set of factoids for a given summary could then be used to evaluate how much information and which content was covered in the summary. Of relevance to this work is that fact that a high level of agreement between the two judges (precision and recall of 96%) was achieved, despite that “very short guidelines” were established for how to identify the factoids. Similarly, in the work by Nenkova and Passonneau [84], assessors labeled Summarization Content Units (SCUs) contained in a given summary. The SCUs are fine-grained, clause-like semantic units (e.g. “two Libyans were indicted” and “in 1991” might be two SCUs in the sentence “Two Libyans were indicted in 1991”). They also note high levels of agreement on annotating the SCUs present in a set of texts between two indepen-

[1] Bahrain television reported 143 people, including 36 children, were on board.
[2] GulfAir said 135 passengers and eight crew members were on board.
[3] There were 135 passengers and eight crew members on board, according to Khaleej Times.
[4] All 143 passengers and crew members were killed.

Figure 5.1: Similar sentence pairs related to the GulfAir plane crash.

dent judges (a Krippendorff's Alpha of 0.81, where values above 0.67 indicate strong reliability [62]). The results of these previous studies are promising evidence that units of text that correspond to a given fact can be annotated reliably.

Another reason I propose the fact-focused sentence-level relevance and novelty task is that a clear criterion for both the labeling of relevant and novel sentences can be stated. Since I assume that the user states the fact of interest to him or her in the form of a question, a relevant sentence should provide an answer to the question. Likewise, a sentence judged as novel should contain a previously unseen answer to the question. I hypothesize that the reliability of novelty judgments should be better in the fact-centered task as compared to finding relevant, novel sentences with respect to a general topic because, as previously noted, novelty is very context-dependent [6]. The related sentences in Figure 5.1, which shows some examples of similar sentences extracted from documents detailing an August 2000 GulfAir plane crash, motivates this stance. The sentences are all relevant to the general topic "August 2000 Gulfair plane crash" and are shown in chronological order with respect to the publication dates of their source news documents.

If we now consider the novelty task, or the process of eliminating the sentences that do not contain novel information, it is not difficult to see why high levels of interjudge agreement are typically not achieved on this task. Sentences 1, 2 and 3 all state the fact that a total of 143 people were on board the plane, while section 4 implies this fact. However, beyond this, the four sentences differ from one another in subtle, yet nontrivial ways:

- Sentence 1 states the number of children on board.
- Sentences 2 and 3 give the number of passengers and crew members on board, but cite different attributions for this information (GulfAir and Khalleej Times, respectively).
- Sentence 4 states that all 143 were killed.

Therefore, one could argue that all four sentences contain some amount of novel information, since they all differ in some way from the others. Still, one might claim that only the first and last sentences report something significantly new, since they have different predicates ("were on board" versus "were killed"). The point is that in the general setting, what "novelty" means is not clear.

Now let us consider the question of novelty in the fact-focus context. Suppose that the question of interest to the user is "*How many people* were on board?" Within this specific context, determining what is novel is more objective. This is because the precise answer to the question is "143," which is expressed in the first sentence. In other words, in the context of this factual question, the latter three sentences do not provide any additional information. However, suppose that the user's question is "*Who* was on board?" In this case, the user is interested in finding descriptions (e.g. names, occupations, ages) of those on board the plane. Therefore, one can conclude that both sentences one and two contain novel information. Sentence one details the total number of passengers and the number of children; sentence two provides the number of passengers and the number of crew members. However, sentences three and four do not provide new information, since they repeat descriptions given in earlier sentences.

In short, since novelty is context-specific, I hypothesize that modifying the sentence-level relevance and novelty task such that it is performed in a specific context, might yield some promising results. Therefore, in the remainder of this paper, I will evaluate the fact-based relevance and novelty task, in order to determine if a reliable data set can be developed for training a system to perform such a task. In addition, in order to have a basis for comparison and discussion of the results, I will also recreate and evaluate the the topic-based task using the same data set.

5.1 Experimental Setup

The annotation experiment was designed to test the following hypotheses regarding the identification of sentences that are relevant to a given information need and that contain novel, previously unseen information:

H1: Annotators will achieve higher levels of agreement in finding sentences relevant to a specific factual question, as compared to finding sentences relevant to a general topic query.

H2: The judges will achieve higher levels of agreement if they are asked to find novel sentences with respect to a factual query, as compared to finding novel sentences in the more general setting.

5.1.1 Data

The data for the experiment came from the 2003 TREC Novelty track test data [116]. The Novelty track clusters consist of 25 news documents (published by three different news agencies) and a general topic (TREC) query. While the 2003 data contained both “event” and “opinion” clusters, I have chosen two of the “event”

```
1. How many people were on board? [number]
2. What was the origin of the plane? [place]
3. Where was the flight's destination? [place]
4. What type of aircraft was the plane? [mark/brand]
5. Where did the plane crash? [place]
6. When did the crash occur? [time]
7. What was the problem with the plane? [reason/cause]
8. Where was the flight data recorder found? [place]
9. How late was the plane taking off in New York? [time duration]
10. How high was the plane flying? [height]
```

Figure 5.2: Factual questions for cluster N4, Egypt Air crash.

clusters that are related to major disasters (an Egypt Air plane crash and the sinking of a Russian submarine). The reason for choosing this subject matter is that such stories have a dynamic element, with the facts surrounding them changing over time, such that being able to identify both relevant and novel information over time is important for understanding them. The attributes for the chosen document clusters are shown in Table 5.1.

Cluster number	**Subject**
N4	Egyptian Air disaster 990
N33	Sinking of Russian submarine Kursk

Table 5.1: Data clusters used in annotation experiment.

I read through all of the documents in each of the two stories, and created a list of ten factual questions that were central to each story. The questions ask about simple yet key facts in the stories, that may change with time as news sources publish additional information, and that expect atomic answers such as a number, name of a person, or a place. The set of 10 questions created for clusters N4 and N33 are shown in Figures 5.2 and 5.3, respectively. In addition, the expected answer types to each question are shown in the square brackets. The corresponding TREC topic queries for the two clusters are shown in Figure 5.4 and Figure 5.5.

1. How many seamen were on the submarine? [number]
2. How was the submarine damaged? [other]
3. What caused the Kursk submarine to sink? [reason/cause]
4. When did the Kursk submarine sink? [date]
5. Where did the Kursk sink? [place]
6. What time did Americans record the sound of an explosion? [time]
7. How far down did the Kursk sink? [depth]
8. Who is the Russian defense minister? [name]
9. Where was Putin during the rescue operation? [place]
10. Which U.S. submarines were in the Barents Sea when the Kursk sank? [name]

Figure 5.3: Factual questions for cluster N33, Sinking of Kursk.

Title:
Egyptian Air disaster 990

Narrative:
Details, technical and otherwise regarding the incident (e.g. number of passengers aboard, number killed, date, time, location, nationalities of victims, crew members, radio contact, radar sightings, rescue efforts and findings) are relevant. Reaction of family members and loved ones regarding the victims are relevant. Investigatory details concerning technical reasons for the crash are relevant. Analysis of recovered items associated with the incident, and the ensuing comments, opinions, findings and reports are relevant. Actions, opinions, and statements from FAA and NTSB, as well as Egyptian CAA personnel regarding the incident including warnings received prior to, and theories concerning the tragedy are relevant. Statements from Machinist Union personnel attesting to the fitness of the plane assembled by their mechanics are relevant.

Description:
Egyptian Air Flight 990 disaster in October of 1999.

Figure 5.4: TREC topic query for cluster N4.

```
Title:
Russian submarine Kursk sinks

Narrative:
Reports on what was known about the sinking of the Russian nuclear
powered submarine, Kursk, are relevant.  Speculation about what
caused the explosions aboard; description of the vessel and its
capabilities, and mention of efforts to rescue the crew are relevant.
Reports that U.S. submarines were monitoring Russian navy exercises
and Russia's suspicions that the Soviet submarine K-128 was struck
by an American submarine and sunk in 1968 are relevant.  Mention of
the fact that Russia turned down a U.S. offer to send a deep-diving
rescue vessel is relevant.  Discussion of U.S. plans to retire one of
its two rescue vessels is not relevant.  Polls reporting how Russians
felt about the disaster and mention of ceremonies for the dead are
relevant.

Description:
The Russian submarine Kursk sank in the Barents Sea killing all 118
aboard in August 2000.
```

Figure 5.5: TREC topic query for cluster N33.

5.1.2 Subjects

Six paid subjects were hired for the experiment. Three were randomly assigned to the test (fact-based) setting and three to the control (topic-based) setting. The experimental design is shown in Table 5.2. In particular, each judge performed the same assigned task on the two Novelty clusters, although in different orders. In both settings, judges were given the 25 news documents for a given cluster, which were numbered from 1 to 25 in chronological order according to their respective publication times. In the control setting, they were also given the TREC topic query for the respective document cluster. To contrast, in the fact-based novelty setting, judges were given the sets of question. The directions for each group (to complete the task on a given cluster of articles) were as follows:

- **Control (topic-based) group:** Familiarize yourself with the story by reading the topic query and by skimming through the news documents. Next, read through the documents carefully in chronological order, recording the document

Judge	Control	Fact-based
A	N33, N4	
B	N33, N4	
C	N4, N33	
D		N4, N33
E		N4, N33
F		N33, N4

Table 5.2: Judges assigned to each setting and the order in which document sets were presented.

number, sentence number and text of each sentence you find that is relevant to the stated topic. When you are finished finding the set of relevant sentences, make a copy of your data. Now, reread through the sentences that you marked as being relevant to the topic, and eliminate those that do not contain novel information. Novel information is "information that has not been previously seen."

- **Test (fact-based) group:** Read through the set of 10 questions. Familiarize yourself with the story by skimming through the set of news documents. Beginning with question one, read through the documents carefully in chronological order, recording the document number, sentence number and text of each sentence you find that provides an answer to the question (a relevant sentence). When you are finished finding the set of relevant sentences, make a copy of your data. Now, reread through the sentences that you marked as being relevant, eliminating those that do not contain novel information. Novel information is "information that has not been previously seen." Use the same procedure for each of the 10 questions.

5.2 Comparison of the Topic and Fact-focused Annotations

I now move on to discussing the results of the annotation experiment. In particular, I will compare the topic and fact-focused annotations with respect to their

Cluster	**Sent.**	**Prop. agree**	**Kappa**
N4	928	0.43	0.20
N33	708	0.33	0.09
Total	1,636	0.39	0.15

Table 5.3: Number of sentences judged, proportion on which all judges agreed, and Kappa for relevance judgments in the control setting.

reproducibility.

5.2.1 Reproducibility of sentence-level relevance judgments

Table 5.3 shows the interjudge agreement between the three annotators in the control setting, who found sentences relevant to the general topic. The table shows for each cluster (as well as over all sentences annotated) the number of sentences, the proportion on which all three judges agreed, and the corresponding Kappa statistic, which factors out the expected (chance) agreement [24][2].

Over both clusters, in which there were a total of 1,636 sentences, all three annotators agreed with respect to whether or not a sentence was relevant on only 39% of the sentences, for a Kappa of 0.15. While there are different scales for interpreting the Kappa statistic (e.g. Landis and Koch [67] state that a Kappa above 0.40 demonstrates a "moderate" degree of interjudge agreement; Krippendorff [62] holds that a Kappa under 0.67 is considered unreliable), a Kappa of 0.15 clearly does not indicate a sufficiently high level of agreement between the three judges.

Table 5.4 shows the level of interjudge agreement between the judges in the test setting, separately by cluster and question, and over the total set of 16,360 sentences annotated. One thing to notice is that agreement varies over the questions. One explanation for this may be that some questions are more difficult to answer than others. For example, there is only one correct answer (namely, "Cairo") that appears

[2] In computing Kappa, I used the method to find the probability of agreement among the annotators due to chance described by Siegel and Castellan[115]. In other words, I assume that there is one probability distribution of the categories relevant/not relevant for all three coders.

in the documents for the question Q3 from cluster N4, “Where was the flight’s destination?” For this question, all three annotators were in perfect agreement as to the set of sentences containing a relevant answer. To contrast, the question from cluster N4 that exhibited the least amount of interjudge agreement was Q7, “What was the problem with the plane?” For this question, there were several relevant answers provided in the various documents, including “no clear mechanical reason why the plane went down” and “left and right elevators were pointing in different directions.” In short, the dynamic nature of the emergency news stories that were used as data can account for some of the difficulty of the annotation task. The more dynamic questions, which ask about facts that are likely to change over time, tend to exhibit less agreement on the relevance judgments.

However, over all of the 16,360 relevance judgments made, the agreement is rather good. All three judges agreed on 99% of the sentences for a Kappa of 0.67. Therefore, I confirm the first hypothesis, that relevance judgments in the fact-based, sentence-level retrieval setting are more reproducible than in the case of the topic-based, control setting.

5.2.2 Reproducibility of sentence-level novelty judgments

I calculated the interjudge agreement on the novelty task with respect to three different sets of sentences. It is defined that all novel sentences must also be relevant, but the judges do not always agree on the relevance judgments. Each of the methods I used has implications for how much the agreement calculation on the novelty judgments is affected by agreement on relevance judgments:

- **All sentences:** A measure of agreement on novelty judgments over all sentences in the document set. Since all sentences are considered, the measure gives credit

Cluster/question	Sent.	Prop. agree	Kappa
N4 Q1	928	0.98	0.70
N4 Q2	928	0.97	0.51
N4 Q3	928	1.00	1.00
N4 Q4	928	0.98	0.63
N4 Q5	928	0.99	0.81
N4 Q6	928	0.98	0.60
N4 Q7	928	0.99	0.20
N4 Q8	928	0.99	0.78
N4 Q9	928	0.99	0.73
N4 Q10	928	0.98	0.59
N33 Q1	708	0.98	0.86
N33 Q2	708	0.98	0.19
N33 Q3	708	0.98	0.24
N33 Q4	708	0.97	0.50
N33 Q5	708	0.98	0.71
N33 Q6	708	0.99	0.88
N33 Q7	708	0.99	0.78
N33 Q8	708	0.99	0.59
N33 Q9	708	0.99	0.69
N33 Q10	708	0.99	0.50
All questions	16,360	0.99	0.67

Table 5.4: Number of sentences judged, proportion on which all judges agreed, and Kappa for relevance judgments in the test setting.

for sentences that are labeled as not novel by all judges, because they were labeled as not being relevant (by all judges) in the first step of the task.

- **The union of the judges' relevant sentences sets:** This considers the agreement on novelty judgments only on the set of sentences labeled as being relevant by at least one judge. Therefore, in cases where there were many disagreements on relevance judgments, these disagreements will also carry over in the novelty judgments.

- **The intersection of the judges' relevant sentences sets:** This uses only the sentences upon which all three judges agree that they are relevant. For cases in which agreement on relevance is low, this calculation may be based on a very small set of sentences, and may therefore not be very robust. In contrast to the first measure based on all sentences, this measure does not give any credit for agreeing on relevance status of sentences - it is based purely on agreement with respect to novelty status. Finally, it is undefined in cases where no sentences are judged as relevant by all judges (or by any judges).

The three-way agreement between judges in the control group, on novelty judgments using the the topic queries, is shown in Table 5.5. The table shows, for each data cluster and measure used (based on all sentences, the union or intersection of the judges' relevant sentence sets), the number of sentences involved, the proportion upon which all three judges agreed on novelty status, and the corresponding Kappa statistic. The agreement over the total set (data clusters N4 and N33) is also shown. Over both clusters N4 and N33, and over all sentences (therefore giving credit for the agreement on non-relevant, non-novel sentences), the Kappa is 0.14. In the most strict case, where I use the union of the judges' relevant sets of sentences, the agreement is below what one would expect by chance (Kappa = -0.06). On the third calculation, over the 184 sentences which all three judges agreed were relevant, the judges agree on novelty status on 73% of the sentences, for a Kappa of 0.54. The first two measures show lower agreement because they are influenced by the low rate of agreement (39%) on the first step - relevance detection. These findings agree with those of Allan and colleagues [7], that relevance detection may be more difficult part of the process.

Tables 5.6, 5.7 and 5.8 show, for each question and cluster combination, the annotation results using the three sets of sentences - all sentences, the union of the judges' relevant sets and the intersection of their relevant sets, respectively. Over all sentences and questions, the judges agreed on 99% of the sentences, for a Kappa of 0.39. To contrast, over the set of sentences for which at least one judge found relevant (the union), the agreement was only 52% (Kappa = 0.18). Finally, on the

Cluster	**Sent.**	**Prop. agree**	**Kappa**
N4-all	928	0.52	0.19
N4-union	649	0.31	0.03
N4-inter	116	0.74	0.63
N33-all	708	0.31	0.05
N33-union	539	0.09	-0.22
N33-inter	68	0.72	-0.10
Both clusters (all)	1,636	0.43	0.14
Both clusters (union)	1,188	0.21	-0.06
Both clusters (inter)	184	0.73	0.54

Table 5.5: Number of sentences judged, proportion on which all judges agreed, and Kappa for novelty judgments in the control setting.

set of sentences that all judges agreed were relevant, the three judges agreed on novelty status in only 46% of the cases (Kappa = 0.27).

On the first two measures of novelty agreement (on the "all" and "union" set of sentences), it is clear that agreement is higher in the fact-based novelty setting as compared to the topic-based setting (Kappa 0.39 vs. 0.14, and 0.18 vs. -0.06, respectively). However, in the case where we consider only the sentences labeled as relevant by all three judges, we find more agreement in the topic-focused case (Kappa of 0.54 vs. 0.27). On the first two measures of agreement on novelty status, it is clear that the fact-based setting benefits from the fact that there is higher agreement on relevance judgments as compared to the topic-focused setting (Kappa of 0.67 vs. 0.15). However, the fact that the topic-based case had better agreement than the fact-focused setting on the third measure is a surprising finding. However, it is clear that we have not achieved enough interjudge agreement for novelty judgments in either setting to conclude that novelty judgments are reproducible. With respect to the second hypothesis, that there is more interjudge agreement on identifying novel sentences given the fact-focused context, as compared to the topical context, I conclude that this is true, when all sentences are considered. Since the task of finding novel sentences is a two-stage process, it seems fair that agreement on the first step,

Cluster/question	Sent.	Prop. agree	Kappa
N4 Q1	928	0.99	0.35
N4 Q2	928	0.98	0.17
N4 Q3	928	0.99	0.25
N4 Q4	928	0.99	0.56
N4 Q5	928	0.98	0.33
N4 Q6	928	0.99	0.18
N4 Q7	928	0.99	0.14
N4 Q8	928	0.99	0.57
N4 Q9	928	0.99	0.87
N4 Q10	928	0.99	0.26
N33 Q1	708	0.97	0.48
N33 Q2	708	0.98	0.22
N33 Q3	708	0.98	0.33
N33 Q4	708	0.99	0.12
N33 Q5	708	1.00	1.00
N33 Q6	708	0.99	0.87
N33 Q7	708	0.99	0.60
N33 Q8	708	1.00	1.00
N33 Q9	708	0.99	0.33
N33 Q10	708	1.00	1.00
All questions (all sentences)	16,360	0.99	0.39

Table 5.6: Number of sentences, proportion on which all judges agreed, and Kappa for novelty judgments in the test setting; all sentences considered.

that of identifying relevant sentences, should not be filtered out when calculating agreement on novelty judgments.

5.3 Discussion

Table 5.9 shows a summary of the interjudge agreement on relevance and novelty status of sentences over the two data clusters, N4 and N33, for both the topic-based (control) setting and the fact-based (test) setting. As previously mentioned, the proposed task involves finding novel sentences using a two-stage process, namely, to first identify relevant sentences and then keep only those containing previously seen information. Therefore, I report the agreement on novelty status over all sentences in the data set. Based on the results, I conclude that there is a satisfactory level of reproducibility on the task of finding sentences that are relevant to factual information needs. However, the reproducibility on the novelty judgments is less

Cluster/question	**Rel. sent.**	**Prop. agree**	**Kappa**
N4 Q1	28	0.71	0.31
N4 Q2	28	0.50	-0.03
N4 Q3	18	0.33	-0.07
N4 Q4	21	0.67	0.41
N4 Q5	25	0.44	0.07
N4 Q6	23	0.44	-0.06
N4 Q7	8	0.25	-0.21
N4 Q8	4	0.25	-0.03
N4 Q9	5	0.80	0.73
N4 Q10	23	0.52	0.06
N33 Q1	24	0.09	-0.26
N33 Q2	17	0.18	-0.20
N33 Q3	19	0.37	0.03
N33 Q4	22	0.68	0.004
N33 Q5	23	1.00	1.00
N33 Q6	3	0.67	-0.13
N33 Q7	6	0.67	0.45
N33 Q8	8	1.00	1.00
N33 Q9	7	0.43	0.067
N33 Q10	4	1.00	1.00
All questions - union	315	0.52	0.18

Table 5.7: Number of relevant sentences (union of judges' relevant sets), proportion on which all judges agreed, and Kappa for novelty judgments in the test setting.

Cluster/question	**Rel. sent.**	**Prop. agree**	**Kappa**
N4 Q1	11	0.64	0.34
N4 Q2	3	0.33	0
N4 Q3	18	0.33	-0.07
N4 Q4	6	0.50	0.25
N4 Q5	14	0.36	0.07
N4 Q6	6	0.17	-0.13
N4 Q7	0	NA	0
N4 Q8	2	0.50	0.25
N4 Q9	2	NA	NA
N4 Q10	7	0.43	0.14
N33 Q1	15	0.07	-0.45
N33 Q2	1	1.00	1.00
N33 Q3	1	1.00	1.00
N33 Q4	3	0.33	0
N33 Q5	9	1.00	1.00
N33 Q6	2	1.00	1.00
N33 Q7	3	1.00	1.00
N33 Q8	1	1.00	1.00
N33 Q9	3	0.33	0.10
N33 Q10	1	1.00	1.00
All questions - intersection	109	0.46	0.27

Table 5.8: Number of relevant sentences (intersection of all judges' relevant sets), proportion on which all judges agreed, and Kappa for novelty judgments in the test setting.

	Judgment	**Prop. agree**	**Kappa**
Topic-based	relevance	0.39	0.15
Fact-based	relevance	0.99	0.67
Topic-based	Novelty (all)	0.43	0.14
Fact-based	Novelty	0.99	0.39

Table 5.9: Summary of differences in interjudge agreement between topic-based and fact-based settings.

promising. While the proportion of sentences upon which the judges agreed is high (99%), the Kappa is only 0.39. In addition, as previously discussed, the agreement for the novelty judgments on the sentences agreed by all judges as being relevant to the given factual question (the "intersection" set), is relatively low - 46% agreement, or a Kappa of only 0.27.

The findings suggest that reliable data sets for system building on the fact-focused, sentence-level relevance detection problem can be produced. However, for the novelty detection problem, it appears that even in the fact-focused task, the concept of what novelty is, is still somewhat subjective.

It should be noted that the problem I have proposed, fact-focused relevance and novelty detection at the sentence level, is closely related to question answering. In fact, the first step, that of identifying sentences relevant to a given factual question, is very similar to passage retrieval for question answering. However, when coupled with the novelty problem, there are some interesting differences. I argue that the proposed problem is more related to that of the Information Synthesis problem [10]. In contrast to the classic Q&A setting, in which the user wants to find the correct answer to a single input question of interest [125], in information synthesis, the user has a more complex information need. For example, in the current problem, I assumed that the user had a set of documents relevant to a general topic, and then he or she created a set of factual questions of interest. In addition, I previously noted that many of the questions in the data were dynamic, in that the answer to them changed over time, which I cited as one reason that some questions yielded lower interjudge agreement on the relevance step than others.

Given the results of the experiment, in future work, it may be more fruitful to concentrate on the detection of relevant sentences (given an input fact) rather than novel ones.

5.4 Conclusion

In this chapter, I proposed the problem of fact-focused relevance and novelty detection at the sentence level. I evaluated its reproducibility, both on the relevance and novelty stages of the task. In addition, I reproduced the TREC Novelty annotation experiment, in which the judges found relevant and novel sentences with respect to a general topic query.

- With respect to finding relevant information at the sentence level, I found that there was greater reproducibility in the fact-focused case as compared to the topic-focused case.

- I showed that interjudge agreement on the novelty judgments can be found in three ways, dependent on the set of sentences one considers. The strictest method is the one that considers novelty agreement with respect to all sentences that at least one judge has labeled as being relevant (the union of relevant sentences). To contrast, which is the most lenient method depends on how much agreement there is on relevance judgments.

- With respect to finding novel information at the sentence level, I found that neither task yielded a satisfactory level of interjudge agreement.

In conclusion, given the results of the experiment, when working at the sentence level of textual granularity, it may be more fruitful to concentrate on the detection of relevant sentences (given an input fact) rather than novel ones. If the goal of developing novelty systems is to help users find relevant information that is not redundant, many system designs are possible that would not rely on automatic novelty detection. For example, sentences relevant to an input fact could be organized in such as way as to help users comprehend the information efficiently (e.g. clustered by keyword, in rank order by relevance). Therefore, in addition to developing effective means to automate the fact-focused sentence retrieval mechanism, future work might also address the design of systems that can facilitate the ease of users detecting new information themselves, given a set of relevant sentences.

In the future, the detection of novel information below the sentence level of textual granularity should be explored. It may be the case that sentences contain too much information for people to make meaningful judgments of novelty at this level. For example, it was illustrated in Chapter III that there are many subtle semantic relationships between a pair of sentences, which both provide an answer to the same question. For example, some of the relationships identified included “partial overlap” (the sentences contain some of the same information, but both also contain something unique) and “attribution” (the sentences provide the same information but one also cites the source of information). In such cases, it may be non-trivial to decide if a sentence provides some significant, novel information, in comparison to previously seen sentences. To contrast, it may be easier to make such comparisons between textual units that describe an atomic fact, as argued in Section 5.0.3. Previous research suggests that users agree on identifying facts in a text [124, 84]. In other words, users agree on fact-level relevance judgments. Therefore, future work should consider if agreement can also be reached on fact-level novelty judgments.

CHAPTER VI

An IR System to Support Short-term Event Tracking

As previously discussed, the central hypothesis of the current thesis is that users can better follow the events described in dynamic, online news stories using an information retrieval system that is specifically designed for this task as compared to using current IR systems. Therefore, I have designed and implemented a prototype system, which I call a "fact tracking system." In the current chapter, the system will be described. In the next chapter, Chapter VII, it will be evaluated in a task oriented user study. In order to test the main hypothesis, the use of the new system will be compared against the use of existing systems.

In Chapter III, I showed that although the answers to factual questions in a corpus of breaking news stories exhibit a good deal of semantic difference over time, commonly used measures of lexical similarity are not correlated to the chronological difference between a pair of answers to a given question. In addition, in the annotation experiment presented in Chapter V, I demonstrated that the sentence-level novelty detection framework (as described by the TREC research initiatives [116]) could not be successfully applied to the problem of automatically detecting sentences in a document set that contain new answers to factual questions over time. In particular, I argued that since human judges do not agree on what sentences are novel, the process cannot be successfully automated in an IR system. However, in contrast to previous work in the TREC domain, I found a satisfactory level of agreement between judges for finding sentences that are relevant to a factual question. Therefore, my system design focuses on the problem of sentence retrieval, given a user's input factual question. Once relevant sentences are retrieved, they may either be sent to an answer extraction module for further processing or they may be presented to the user as a question-focused summary. Regardless of the output desired by the user, the relevant information is ordered chronologically and is presented to the user with its respective source and time of publication.

Figure 6.1 shows the architecture of the fact tracking system, including the answer extraction component. As can be seen, the input to the system is a set of news articles related to a breaking news story of interest, as well as a factual question to track over time and source. The system is made up of three components - a question-focused version of the MEAD summarizer that I have tuned for the dynamic text sentence retrieval problem, a modified version of the NSIR question answering system that extracts answers from the sentences previously identified by MEAD, and a graphical representation of the extracted answers, shown by their respective publication times and sources.

While the focus of the thesis is the evaluation of the system design, rather than the optimization of the overall system performance, I conducted a set of passage retrieval experiments in order to tune the MEAD summarizer for this task, when given a set of dynamic, breaking news stories. Therefore, the next section (Section 6.1) will describe these experiments in detail. Next, the remainder of the chapter will explain how the three components were integrated in creating the fact tracking system.

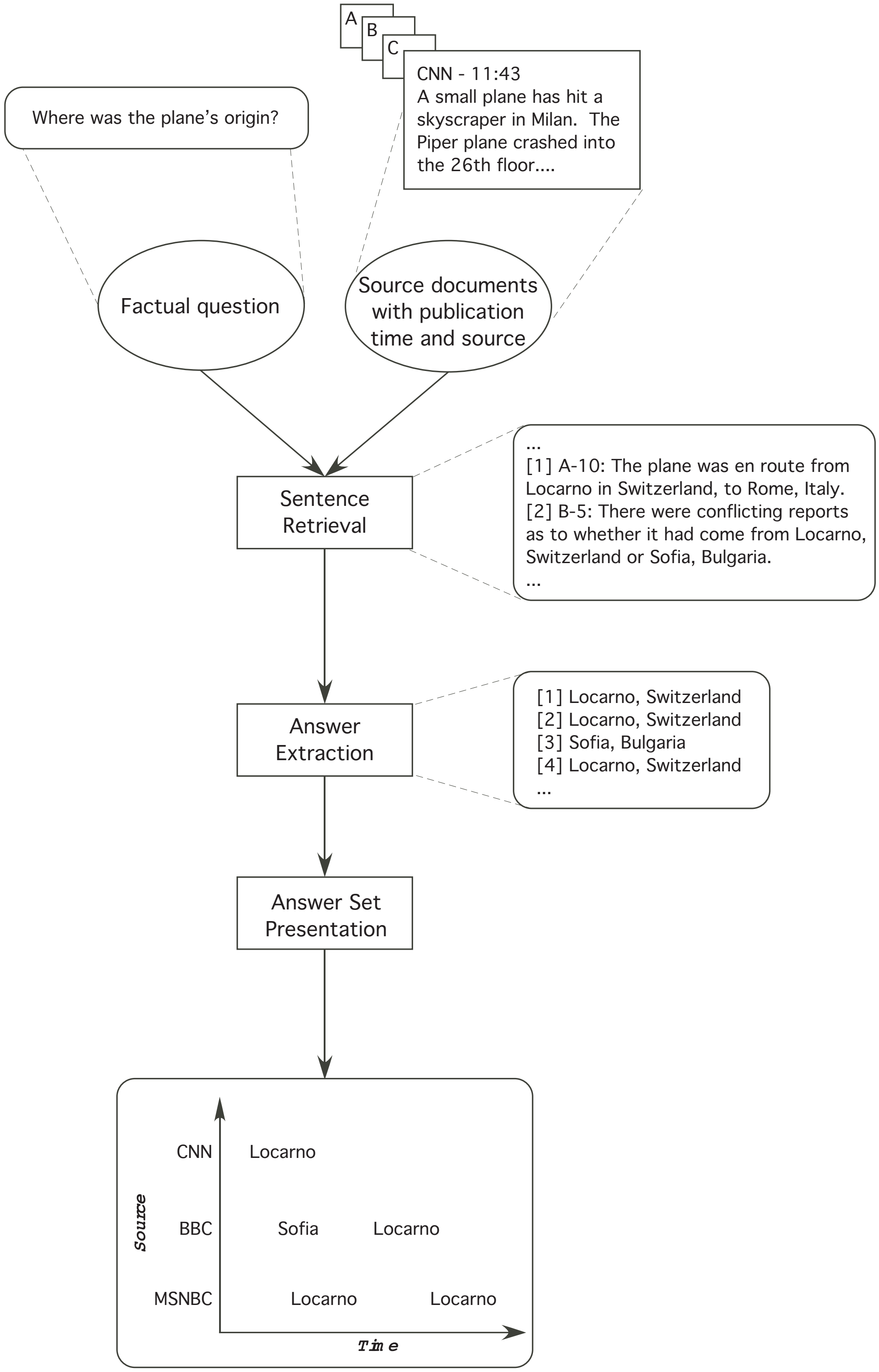
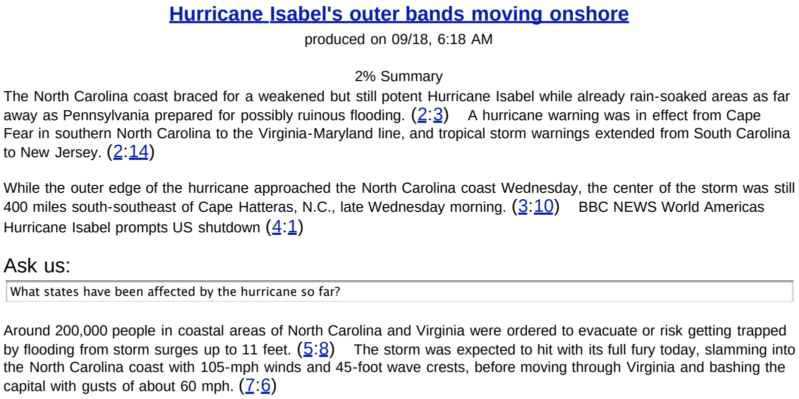

Where was the plane's origin?
A
B
C
CNN - 11:43
A small plane has hit a skyscraper in Milan. The Piper plane crashed into the 26th floor....
Factual question
Source documents with publication time and source
Sentence Retrieval
...
[1] A-10: The plane was en route from Locarno in Switzerland, to Rome, Italy.
[2] B-5: There were conflicting reports as to whether it had come from Locarno, Switzerland or Sofia, Bulgaria.
...
Answer Extraction
[1] Locarno, Switzerland
[2] Locarno, Switzerland
[3] Sofia, Bulgaria
[4] Locarno, Switzerland
...
Answer Set Presentation
Source
CNN
BBC
MSNBC
Locarno
Sofia
Locarno
Locarno
Locarno
Time

Figure 6.1: Fact Tracking System Architecture

6.1 Passage Retrieval with the MEAD Summarizer

Recent work has motivated the need for systems that support "Information Synthesis" tasks, in which a user seeks a global understanding of a topic or story [10]. In contrast to the classical question answering setting (e.g. TREC-style Q&A [125]), in which the user presents a single question and the system returns a corresponding answer (or a set of likely answers), in this case the user has a more complex information need.

Similarly, when reading about a complex news story, such as an emergency situation, users might seek answers to a set of questions in order to understand it better. For example, Figure 6.2 shows the interface to a Web-based news summarization system, which a user has queried for information about Hurricane Isabel. Understanding such stories is challenging for a number of reasons. In particular, complex stories contain many sub-events (e.g. the devastation of the hurricane, the relief effort, etc.) In addition, while some facts surrounding the situation do not change (such as "Which area did the hurricane first hit?"), others may change with time ("How many people have been left homeless?").



Hurricane Isabel's outer bands moving onshore

produced on 09/18, 6:18 AM

2% Summary

The North Carolina coast braced for a weakened but still potent Hurricane Isabel while already rain-soaked areas as far away as Pennsylvania prepared for possibly ruinous flooding. (2:3) A hurricane warning was in effect from Cape Fear in southern North Carolina to the Virginia-Maryland line, and tropical storm warnings extended from South Carolina to New Jersey. (2:14)

While the outer edge of the hurricane approached the North Carolina coast Wednesday, the center of the storm was still 400 miles south-southeast of Cape Hatteras, N.C., late Wednesday morning. (3:10) BBC NEWS World Americas Hurricane Isabel prompts US shutdown (4:1)

Ask us:

What states have been affected by the hurricane so far?

Around 200,000 people in coastal areas of North Carolina and Virginia were ordered to evacuate or risk getting trapped by flooding from storm surges up to 11 feet. (5:8) The storm was expected to hit with its full fury today, slamming into the North Carolina coast with 105-mph winds and 45-foot wave crests, before moving through Virginia and bashing the capital with gusts of about 60 mph. (7:6)

Figure 6.2: Question tracking interface to a summarization system.

Currently, I address the question-focused sentence retrieval task. While passage retrieval (PR) is clearly not a new problem (e.g. [108, 109]), it remains important and yet often overlooked. As noted by [45], while PR is the crucial first step for question answering, Q&A research has typically not emphasized it. The specific problem I consider differs from the classic task of PR for a Q&A system in interesting ways, due to the time-sensitive nature of the stories in the corpus. For example, one challenge is that the answer to a user's question may be updated and reworded over time by journalists in order to keep a running story fresh, or because the facts themselves change. Therefore, there is often more than one correct answer to a question.

The current aim is to develop a method for sentence retrieval that goes beyond finding sentences that are similar to a single query. To this end, I propose to use a stochastic, graph-based method. Recently, graph-based methods have proved useful for a number of NLP and IR tasks such as document re-ranking in ad hoc IR [65] and analyzing sentiments in text [91]. In [38], the LexRank method was introduced and was successfully applied it to generic, multi-document summarization. Presently, a topic-sensitive LexRank is developed, in creating a sentence retrieval module. I will then evaluate its performance against a competitive baseline, which considers the similarity between each sentence and the question (using IDF-weighed word overlap). I will demonstrate that LexRank significantly improves question-focused sentence selection over the baseline.

6.1.1 Description of the problem

The goal is to build a question-focused sentence retrieval mechanism using a topic-sensitive version of the LexRank method. In contrast to previous PR systems such as Okapi [108], which ranks documents for relevance and then proceeds to find paragraphs related to a question, I address the finer-grained problem of finding sentences containing answers. In addition, the input to the system is a set of documents relevant to the topic of the query that the user has already identified (e.g. via a search engine). The system does not rank the input documents, nor is it restricted in terms of the number of sentences that may be selected from the same document.

The output of the system, a ranked list of sentences relevant to the user's question, can be subsequently used as input to an answer selection system in order to find specific answers from the extracted sentences. Alternatively, the sentences can be returned to the user as a question-focused summary. This is similar to "snippet retrieval" [128]. However, in the current system answers are extracted from a set of multiple documents rather than on a document-by-document basis.

6.1.2 The new approach: topic-sensitive LexRank

In [38], the concept of graph-based centrality was used to rank a set of sentences, in producing generic multi-document summaries. To apply LexRank, a similarity graph is produced for the sentences in an input document set. In the graph, each node represents a sentence. There are edges between nodes for which the cosine similarity between the respective pair of sentences exceeds a given threshold. The degree of a given node is an indication of how much information the respective sentence has in common with other sentences. Therefore, sentences that contain the most salient information in the document set should be very central within the graph.

Figure 6.3 shows an example of a similarity graph for a set of five input sentences, using a cosine similarity threshold of 0.15. Once the similarity graph is constructed, the sentences are then ranked according to their eigenvector centrality. As previously mentioned, the original LexRank method performed well in the context of generic summarization. Below, I describe a topic-sensitive version of LexRank, which is more appropriate for the question-focused sentence retrieval problem. In the new approach, the score of a sentence is determined by a mixture model of the relevance of the sentence to the query and the similarity of the sentence to other high-scoring sentences.

Relevance to the question

In topic-sensitive LexRank, all sentences in a set of articles are stemmed and the word IDFs are computed by the following formula:

$$\text{idf}_w = \log\Big(\frac{N+1}{0.5 + sf_w}\Big) \tag{6.1}$$

where N is the total number of sentences in the cluster, and sf_w is the number of sentences that the word w appears in.

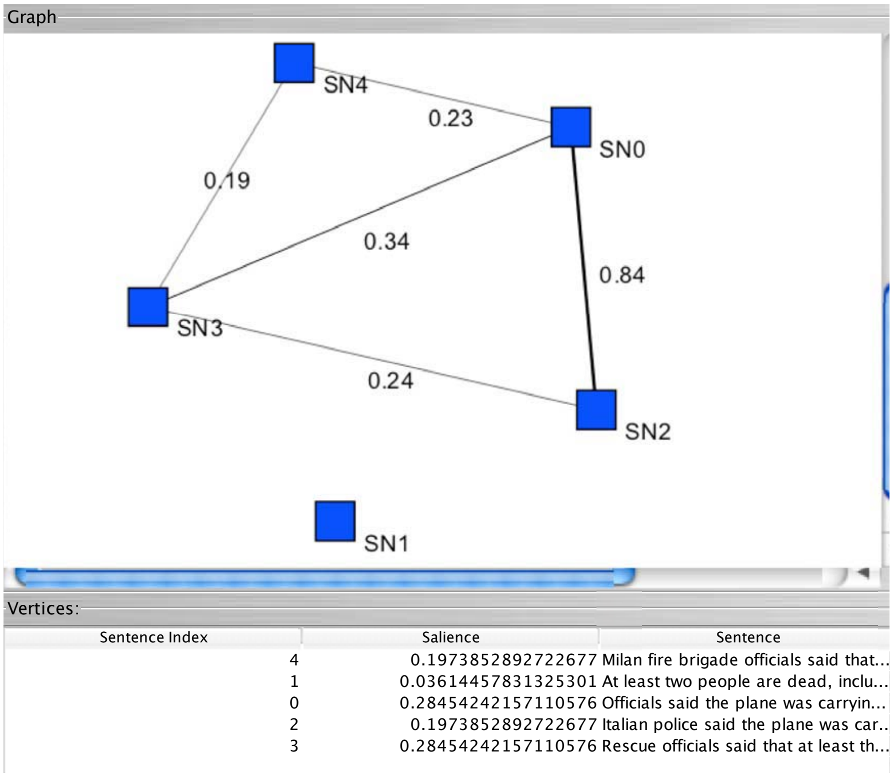
The question is also stemmed, and the stop words are removed from it. Then the relevance of a sentence s to the question q is computed by:

$$\text{rel}(s|q) = \sum_{w \in q} \log(tf_{w,s} + 1) \times \log(tf_{w,q} + 1) \times \text{idf}_w \tag{6.2}$$

where $tf_{w,s}$ and $tf_{w,q}$ are the number of times w appears in s and q, respectively. This model has proven to be successful in query-based sentence retrieval [8], and is used as the competitive baseline in this study (e.g. Tables 6.4, 6.5 and 6.7).

The mixture model

The baseline system explained above does not make use of any inter-sentence information in a cluster. I hypothesize that a sentence that is similar to the high scoring sentences in the cluster should also have a high score. For instance, if a sentence that gets a high score in the baseline model is likely to contain an answer to the question, then a related sentence, which may not be similar to the question itself, is also likely to contain an answer.



Sentence Index	Salience	Sentence
4	0.1973852892722677	Milan fire brigade officials said that...
1	0.03614457831325301	At least two people are dead, inclu...
0	0.28454242157110576	Officials said the plane was carryin...
2	0.1973852892722677	Italian police said the plane was car..
3	0.28454242157110576	Rescue officials said that at least th...

Figure 6.3: LexRank example: sentence similarity graph with a cosine threshold of 0.15.

This idea is captured by the following mixture model, where $p(s|q)$, the score of a sentence s given a question q, is determined as the sum of its relevance to the question (using the same measure as the baseline described above) and the similarity to the other sentences in the document cluster:

$$p(s|q) = d\frac{\text{rel}(s|q)}{\sum_{z\in C}\text{rel}(z|q)} + (1-d)\sum_{v\in C}\frac{\text{sim}(s,v)}{\sum_{z\in C} sim(z,v)}p(v|q) \tag{6.3}$$

where C is the set of all sentences in the cluster. The value of d, which will also be referred to as the "question bias," is a trade-off between two terms in the equation and is determined empirically. For higher values of d, more importance is given to the relevance to the question compared to the similarity to the other sentences in the cluster. The denominators in both terms are for normalization, which are described below. The cosine measure weighted by word IDFs is used as the similarity between two sentences in a cluster:

$$\mathrm{sim}(x,y) = \frac{\sum_{w \in x,y} \mathrm{tf}_{w,x}\mathrm{tf}_{w,y}(\mathrm{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\mathrm{tf}_{x_i,x}\mathrm{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\mathrm{tf}_{y_i,y}\mathrm{idf}_{y_i})^2}} \tag{6.4}$$

Equation 6.3 can be written in matrix notation as follows:

$$\mathbf{p} = [d\mathbf{A} + (1-d)\mathbf{B}]^{\mathrm{T}}\mathbf{p} \tag{6.5}$$

$\mathbf{A}$ is the square matrix such that for a given index i, all the elements in the i^{th} column are proportional to $\mathrm{rel}(i|q)$. $\mathbf{B}$ is also a square matrix such that each entry $\mathbf{B}(i,j)$ is proportional to $sim(i,j)$. Both matrices are normalized so that row sums add up to 1. Note that as a result of this normalization, all rows of the resulting square matrix $\mathbf{Q} = [d\mathbf{A} + (1-d)\mathbf{B}]$ also add up to 1. Such a matrix is called *stochastic* and defines a Markov chain. If each sentence is viewed as a state in a Markov chain, then $\mathbf{Q}(i,j)$ specifies the transition probability from state i to state j in the corresponding Markov chain. The vector $\mathbf{p}$ we are looking for in Equation 6.5 is the stationary distribution of the Markov chain. An intuitive interpretation of the stationary distribution can be understood by the concept of a random walk on the graph representation of the Markov chain.

With probability d, a transition is made from the current node (sentence) to the nodes that are similar to the query. With probability (1-d), a transition is made to the nodes that are lexically similar to the current node. Every transition is weighted according to the similarity distributions. Each element of the vector $\mathbf{p}$ gives the asymptotic probability of ending up at the corresponding state in the long run regardless of the starting state. The stationary distribution of a Markov chain can be computed by a simple iterative algorithm, called power method.[1]

[1] The stationary distribution is unique and the power method is guaranteed to converge provided that the Markov chain is ergodic [114]. A non-ergodic Markov chain can be made ergodic by reserving a small probability for jumping to any other state from the current state [89].

A simpler version of Equation 6.5, where **A** is a uniform matrix and **B** is a normalized binary matrix, is known as PageRank [19, 89] and used to rank the web pages by the Google search engine. It was also the model used to rank sentences in [38].

Experiments with topic-sensitive LexRank

I experimented with different values of d on the training data. I also considered several threshold values for inter-sentence cosine similarities, where I ignored the similarities between the sentences that are below the threshold. In the training phase of the experiment, I evaluated all combinations of LexRank with d in the range of $[0, 1]$ (in increments of 0.10) and with a similarity threshold ranging from $[0, 0.9]$ (in increments of 0.05). I then found all configurations that outperformed the baseline. These configurations were then applied to the development/test set. Finally, the best sentence retrieval system was applied to the test data set and evaluated against the baseline. The remainder of the paper will explain this process and the results in detail.

6.1.3 Experimental setup

I built a corpus of 20 multi-document clusters of complex news stories, such as plane crashes, political controversies and natural disasters. The data clusters and their characteristics are shown in Table 6.1. The number of clusters randomly assigned to the training, development/test and test data sets were 11, 3 and 6, respectively.

Next, I assigned each cluster of articles to an annotator, who was asked to read all articles in the cluster. He or she then generated a list of factual questions key to understanding the story. Once I collected the questions for each cluster, two judges independently annotated nine of the training clusters. For each sentence and question pair in a given cluster, the judges were asked to indicate whether or not the sentence contained a complete answer to the question. Once an acceptable rate of interjudge agreement was verified on the first nine clusters (Kappa [24] of 0.68), the remaining 11 clusters were annotated by one judge each.

In some cases, the judges did not find any sentences containing the answer for a given question. Such questions were removed from the corpus. The final number of questions annotated for answers over the entire corpus was 341, and the distributions of questions per cluster can be found in Table 6.1.

Evaluation metrics and methods

To evaluate the sentence retrieval mechanism, I produced extract files, which contain a list of sentences deemed to be relevant to the question, for the system and from human judgment. To compare different configurations of the system to the baseline system, I produced extracts at a fixed length of 20 sentences. While evaluations of question answering systems are often based on a shorter list of ranked sentences, I chose to generate longer lists for several reasons. One is that I am developing a PR module, of which the output can then be input to an answer extraction system for further processing. In such a setting, we would most likely want to generate a relatively longer list of candidate sentences. As previously mentioned, in the corpus the questions often have more than one relevant answer, so ideally, the PR system would find many of the relevant sentences, sending them on to the answer component to decide which answer(s) should be returned to the user. Each system's extract file lists the document and sentence numbers of the top 20 sentences. The "gold standard" extracts list the sentences judged as containing answers to a given question by

Story	**Documents**	**Questions**	**Data set**	**Sample question**
Algerian terror threat	2	12	train	What is the condition under which GIA will take action?
Milan plane crash	9	15	train	How many people were in the building at the time?
Turkish plane crash	10	12	train	To where was the plane headed?
Moscow terror attack	7	7	train	How many people were killed in the recent explosion?
Rhode Island club fire	10	8	train	Who was to blame for the fire?
FBI most wanted	3	14	train	How much is the State Department offering for information leading to bin Laden's arrest?
Russia bombing	2	11	train	What was the cause of the blast?
Bali terror attack	10	30	train	What were the motives of the attackers?
DC sniper	8	28	train	What kinds of weapons or equipment were used?
GSPC terror group	8	29	train	What are the charges against the GSPC suspects?
China earthquake	25	18	train	What was the magnitude of the quake in Zhangjiakou?
Gulfair plane crash	11	29	dev/test	How many people were on board?
David Beckham trade	20	28	dev/test	How long had Beckham been playing for MU before he moved to RM?
Miami airport evacuation	12	15	dev/test	How many concourses does the airport have?
US hurricane	14	14	test	In which places had the hurricane landed?
EgyptAir crash	25	29	test	How many people were killed?
Kursk submarine disaster	25	30	test	When did the Kursk sink?
Hebrew University bombing	11	27	test	How many people were injured?
Finland mall bombing	9	15	test	How many people were in the mall at the time?
Putin visits England	12	20	test	What issue concerned British human rights groups?

Table 6.1: Corpus of breaking news stories.

the annotators (and therefore have variable sizes) in no particular order.[2]

I evaluated the performance of the systems using two metrics - Mean Reciprocal Rank (MRR) [125] and Total Reciprocal Document Rank (TRDR) [98]. MRR, used in the TREC Q&A evaluations, is the reciprocal rank of the first correct answer (or sentence, in this case) to a given question. This measure gives us an idea of how far down we must look in the ranked list in order to find a correct answer. To contrast, TRDR is the total of the reciprocal ranks of all answers found by the system. In the context of answering questions from complex stories, where there is often more than one correct answer to a question, and where answers are typically time-dependent, I should focus on maximizing TRDR, which gives us a measure of how many of the relevant sentences were identified by the system. However, I report both the average MRR and TRDR over all questions in a given data set.

6.1.4 LexRank versus the baseline system

In the training phase, I searched the parameter space for the values of d (the question bias) and the similarity threshold in order to optimize the resulting TRDR scores. For the current problem, I expected that a relatively low similarity threshold pair with a high question bias would achieve the best results. Table 6.2 shows the effect of varying the similarity threshold.[3] The notation $LR[a, d]$ is used, where a is the similarity threshold and d is the question bias. The optimal range for the parameter a was between 0.14 and 0.20. This is intuitive because if the threshold is too high, such that only the most lexically similar sentences are represented in the graph, the method does not find sentences that are related but are more lexically diverse (e.g. paraphrases). Table 6.3 shows the effect of varying the question bias at two different similarity thresholds (0.02 and 0.20). It is clear that a high question

[2] For clusters annotated by two judges, all sentences chosen by at least one judge were included.

[3] A threshold of -1 means that no threshold was used such that all sentences were included in the graph.

System	**Ave. MRR**	**Ave. TRDR**
LR[-1.0,0.65]	0.5270	0.8117
LR[0.02,0.65]	0.5261	0.7950
LR[0.16,0.65]	0.5131	0.8134
LR[0.18,0.65]	0.5062	0.8020
LR[0.20,0.65]	0.5091	0.7944
LR[-1.0,0.80]	0.5288	0.8152
LR[0.02,0.80]	0.5324	0.8043
LR[0.16,0.80]	0.5184	0.8160
LR[0.18,0.80]	0.5199	0.8154
LR[0.20,0.80]	0.5282	0.8152

Table 6.2: Training phase: effect of similarity threshold (a) on Ave. MRR and TRDR.

System	**Ave. MRR**	**Ave. TRDR**
LR[0.02,0.65]	0.5261	0.7950
LR[0.02,0.70]	0.5290	0.7997
LR[0.02,0.75]	0.5299	0.8013
LR[0.02,0.80]	0.5324	0.8043
LR[0.02,0.85]	0.5322	0.8038
LR[0.02,0.90]	0.5323	0.8077
LR[0.20,0.65]	0.5091	0.7944
LR[0.20,0.70]	0.5244	0.8105
LR[0.20,0.75]	0.5285	0.8137
LR[0.20,0.80]	0.5282	0.8152
LR[0.20,0.85]	0.5317	0.8203
LR[0.20,0.90]	0.5368	0.8265

Table 6.3: Training phase: effect of question bias (d) on Ave. MRR and TRDR.

bias is needed. However, a small probability for jumping to a node that is lexically similar to the given sentence (rather than the question itself) is needed. Table 6.4 shows the configurations of LexRank that performed better than the baseline system on the training data, based on mean TRDR scores over the 184 training questions. I applied all four of these configurations to the unseen development/test data, in order to see if I could further differentiate their performances.

Development/testing phase

The scores for the four LexRank systems and the baseline on the development/test data are shown in Table 6.5. This time, all four LexRank systems outperformed the baseline, both in terms of average MRR and TRDR scores. An analysis of the average

System	Ave. MRR	Ave. TRDR
Baseline	0.5518	0.8297
LR[0.14,0.95]	0.5267	0.8305
LR[0.18,0.90]	0.5376	0.8382
LR[0.18,0.95]	0.5421	0.8382
LR[0.20,0.95]	0.5404	0.8311

Table 6.4: Training phase: systems outperforming the baseline in terms of TRDR score.

System	Ave. MRR	Ave. TRDR
Baseline	0.5709	1.0002
LR[0.14,0.95]	0.5882	1.0469
LR[0.18,0.90]	0.5820	1.0288
LR[0.18,0.95]	0.5956	1.0411
LR[0.20,0.95]	0.6068	1.0601

Table 6.5: Development testing evaluation.

scores over the 72 questions within each of the three clusters for the best system, LR[0.20,0.95], is shown in Table 6.6. While LexRank outperforms the baseline system on the first two clusters both in terms of MRR and TRDR, their performances are not substantially different on the third cluster. Therefore, I examined properties of the questions within each cluster in order to see what effect they might have on system performance.

I hypothesized that the baseline system, which compares the similarity of each sentence to the question using IDF-weighted word overlap, should perform well on questions that provide many content words. To contrast, LexRank might perform better when the question provides fewer content words, since it considers both similarity to the query and inter-sentence similarity. Out of the 72 questions in the

Cluster	B-MRR	LR-MRR	B-TRDR	LR-TRDR
Gulfair	0.5446	0.5461	0.9116	0.9797
David Beckham trade	0.5074	0.5919	0.7088	0.7991
Miami airport evacuation	0.7401	0.7517	1.7157	1.7028

Table 6.6: Average scores by cluster: baseline versus LR[0.20,0.95].

	Ave. MRR	**Ave. TRDR**
Baseline	0.5780	0.8673
LR[0.20,0.95]	0.6189	0.9906
p-value	na	0.0619

Table 6.7: Testing phase: baseline vs. LR[0.20,0.95].

development/test set, the baseline system outperformed LexRank on 22 of the questions. In fact, the average number of content words among these 22 questions was slightly, but not significantly, higher than the average on the remaining questions (3.63 words per question versus 3.46). Given this observation, I experimented with two mixed strategies, in which the number of content words in a question determined whether LexRank or the baseline system was used for sentence retrieval. I tried threshold values of 4 and 6 content words, however, this did not improve the performance over the pure strategy of system LR[0.20,0.95]. Therefore, I applied this system versus the baseline to the unseen test set of 134 questions.

Testing phase

As shown in Table 6.7, LR[0.20,0.95] outperformed the baseline system on the test data both in terms of average MRR and TRDR scores. The improvement in average TRDR score was statistically significant with a p-value of 0.0619. Since I am interested in a passage retrieval mechanism that finds sentences relevant to a given question, providing input to the question answering component of the system, the improvement in average TRDR score is very promising. While we saw in Section 6.1.4 that LR[0.20,0.95] may perform better on some question or cluster types than others, I conclude that it beats the competitive baseline when one is looking to optimize mean TRDR scores over a large set of questions.

6.1.5 Discussion

The idea behind using LexRank for sentence retrieval is that a system that considers only the similarity between candidate sentences and the input query, and not the similarity between the candidate sentences themselves, is likely to miss some important sentences. When using any metric to compare sentences and a query, there is always likely to be a tie between multiple sentences (or, similarly, there may be cases where fewer than the number of desired sentences have similarity scores above zero). LexRank effectively provides a means to break such ties. An example of such a scenario is illustrated in Tables 6.8 and 6.9, which show the top ranked sentences by the baseline and LexRank, respectively for the question "What caused the Kursk to sink?" from the Kursk submarine cluster. It can be seen that all top five sentences chosen by the baseline system have the same sentence score (similarity to the query), yet the top ranking two sentences are not actually relevant according to the judges. To contrast, LexRank achieved a better ranking of the sentences since it is better able to differentiate between them. It should be noted that both for the LexRank and baseline systems, chronological ordering of the documents and sentences is preserved, such that in cases where two sentences have the same score, the one published earlier is ranked higher.

In addition to improving the question-focused sentence retrieval performance of biased LexRank in future work, other classification algorithms might also be tested on this task. For example, Radev[96] has been developing methods of weakly supervised graph-based algorithms, which could easily be applied to the problem of ranking sentences with respect to a question of interest.

Rank	Sentence	Score	Relevant?
1	The Russian governmental commission on the accident of the submarine Kursk sinking in the Barents Sea on August 12 has rejected 11 original explanations for the disaster, but still cannot conclude what caused the tragedy indeed, Russian Deputy Premier Ilya Klebanov said here Friday.	4.2282	N
2	There has been no final word on what caused the submarine to sink while participating in a major naval exercise, but Defense Minister Igor Sergeyev said the theory that Kursk may have collided with another object is receiving increasingly concrete confirmation.	4.2282	N
3	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	4.2282	Y
4	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	4.2282	Y
5	President Clinton's national security adviser, Samuel Berger, has provided his Russian counterpart with a written summary of what U.S. naval and intelligence officials believe caused the nuclear-powered submarine Kursk to sink last month in the Barents Sea, officials said Wednesday.	4.2282	N

Table 6.8: Top ranked sentences using baseline system on the question “What caused the Kursk to sink?”.

Rank	Sentence	Score	Relevant?
1	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	0.0133	Y
2	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	0.0133	Y
3	The Russian navy refused to confirm this, but officers have said an explosion in the torpedo compartment at the front of the submarine apparently caused the Kursk to sink.	0.0125	Y
4	President Clinton's national security adviser, Samuel Berger, has provided his Russian counterpart with a written summary of what U.S. naval and intelligence officials believe caused the nuclear-powered submarine Kursk to sink last month in the Barents Sea, officials said Wednesday.	0.0124	N
5	There has been no final word on what caused the submarine to sink while participating in a major naval exercise, but Defense Minister Igor Sergeyev said the theory that Kursk may have collided with another object is receiving increasingly concrete confirmation.	0.0123	N

Table 6.9: Top ranked sentences using the LR[0.20,0.95] system on the question "What caused the Kursk to sink?"

6.2 Answer Extraction with the NSIR System

As mentioned, the passage retrieval component was designed to return the top-ranking 20 sentences, or the 20 sentences that are most likely to contain a relevant answer to the input question. These 20 candidate sentences are then sent to an answer extraction module for further processing, in order to find the specific lexical items that correspond to the answer to the question. For the answer extraction component, I use NSIR [98], a state-of-the-art question answering system. NSIR is an open-domain system that takes a user's question as input, and uses documents on the Web to find answers to the question. In addition, NSIR runs on the Web in real time.

In my problem, I assume that the user has identified a set of news articles that are relevant to his or her topic of interest (e.g. using a search engine or a news delivery service). In addition, I use the sentence retrieval mechanism described in the previous section, which was designed specifically for the purpose of finding sentences in a set of breaking news stories that are relevant to an input factual question. Therefore, I use only the answer extraction components of NSIR, which make up the "probabilistic phrase reranking" method. While [98] provides a detailed description and evaluation of this method, here I give a brief overview of the steps that are involved. Given a list of sentences that are likely to be relevant to a user's question of interest, NSIR performs the following steps:

1. **Question-type recognition**: NSIR uses a rule-based classifier to determine the question's type. For example, question types might be "person," "place" or "distance."

2. **Potential answer extraction**: Each of the input sentences is split into its

constituent phrases, each of which is a potential answer candidate.

3. **Answer ranking**: Each candidate answer is assigned a probability of being the correct answer. This probability score is based on two features - the proximity of the candidate answer to query (question) terms in the original question, as well as the likelihood of the phrase corresponding to the answer, given the question's respective type.

I use NSIR to extract and rank the potential answers contained in a given set of relevant sentences. I then output the top 10 candidate answers. Since ideally, NSIR should be further processing the list of 20 relevant sentences, narrowing down the number of items that are relevant to the question, it was decided to return 10 answers back to the users. This reduces the number of items returned to the user by half. However, 10 candidate answers may appear to be a large number, as compared to typical evaluations of Q&A systems. Since in the current work, the focus was on tuning the passage retrieval component of the system rather than on improving the answer extraction module, there is still room for much improvement on answer ranking. Therefore, I wanted to ensure that enough quality answers were still being returned to the user for the experiments described in the next chapter. As mentioned, returning a list of 10 answers reduced the output seen by the user by half (as compared to the list of 20 relevant passages), while at the same time catching enough of the correct answers.

CHAPTER VII

Short-term Event Tracking: a User Study

This chapter describes a user study that was undertaken in order to evaluate the effectiveness of the prototype IR system that was described in Chapter VI. As previously mentioned, while the intrinsic performance of the system itself can certainly be improved in future work, the goal of the current study is to test the hypothesis that a system that is specifically designed to support the problem of following changing events in a breaking news story, by presenting relevant information by publication time and source, can more effectively help users as compared to existing IR systems that do not take considerations of source and time into account. Another goal of the study is to evaluate some of the possible benefits and drawbacks of the current, very basic system implementation for future improvements. In other words, this chapter describes an initial investigation to see if the general approach and initial system design are promising. Finally, the experiment should also give us a better understanding of the task of finding facts in time-dependent, textual information, as well as what modifications of the current design users might find most beneficial.

7.1 Study Setup

38 subjects were recruited for the study. The subjects were recruited through an email announcement that was sent to all graduate students in the School of

Information and all were self-described native or near-native speakers of English. Since the target users of the system are technically savvy individuals who are likely to seek time-sensitive information from the Web, it was thought that this subject pool was the most appropriate. The subjects made appointments for the experiment on a first-come, first-serve basis. The first five people who participated were treated as pilot subjects and were used in order to validate the study design and tasks. Therefore, data from these subjects are not reported in the results. The subjects were paid a flat rate of $20 for their participation in the study, which on average, took one hour to complete.

7.1.1 Pilot studies

As mentioned, five pilot user studies were run before beginning to collect experimental data. The pilot studies were carried out using the study design and protocol and the subjects were not aware that they were pilot subjects. While there were no problems with the study tasks, it became clear that the manner in which the system output was displayed was problematic. As described in Chapter VI and depicted in Figure 6.1, originally, the intent was to display the relevant answers (or relevant sentences) to the input question that were found by the system in a graph, by their respective publication times and sources. However, in the exit interviews of the pilot studies, it became clear that not only were the graphical displays not helpful to the subjects, they were confusing. When asked for changes they would like to see made to the systems, four of the five subjects directly stated that the graphical display should be removed. Therefore, the graphical depiction of the extracted answers to questions was not included with the system output in the current user study. This also has the added benefit that the output of all three systems tested in the user study could be displayed in tables in simple web pages, and essentially looked the

Breaking News Story	**# News sources**	**Time span**	**# Articles**
Milan plane crash	5	30 hours	56
RI nightclub fire	9	22 hours	43
Sinking of Kursk submarine	3	28 days	25

Table 7.1: Breaking news stories in user study.

same. Therefore, the focus of the study was on evaluating the content of the systems' output, rather than how the content was visualized.

7.1.2 Experimental design

All subjects completed three information-seeking tasks that each involved answering five factual questions about a set of news articles related to a breaking story. This task is similar to the information synthesis task, in which a user wants to get answers to a set of questions in order to understand a given topic or story better [10]. However, in the current study, users were assigned 5 questions that were deemed as being key to understanding the story. This is because, in this initial study, the goal was to evaluate how well the system does at helping users to answer time and source-sensitive questions. Therefore, each task consisted of answering five questions whose answers were known to vary across news articles. Each subject completed each of the three tasks using the output of a different information retrieval system. In addition, in all three experimental settings, the subjects were given the original news articles as well as their respective publication time and source information. The three breaking news stories that were used in the study are described in Table 7.1, while the specific tasks the users completed will be detailed in Section 7.1.3.

Since 33 test subjects were used in the study, each task and system were paired at least three times. In addition, in order to control for any learning effects, the order in which subjects encountered the systems and tasks was varied using the experimental design shown in Table 7.2. In the table, the notation "$X - \#$" is used, in which

User ID	1st task	2nd task	3rd task
1	M1	R2	K3
2	M1	K3	R2
3	R2	M1	K3
4	R2	K3	M1
5	K3	R2	M1
6	K3	M1	R2
7	M1	K2	R3
8	M1	R3	K2
9	K2	M1	R3
10	K2	R3	M1
11	R3	M1	K2
12	R3	K2	M1
13	R1	M2	K3
14	R1	K3	M2
15	M2	R1	K3
16	M2	K3	R1
17	K3	R1	M2
18	K3	M2	R1
19	R1	K2	M3
20	R1	M3	K2
21	K2	R1	M3
22	K2	M3	R1
23	M3	K2	R1
24	M3	R1	K2
25	K1	R2	M3
26	K1	M3	R2
27	R2	K1	M3
28	R2	M3	K1
29	M3	R2	K1
30	M3	K1	R2
31	K1	M2	R3
32	K1	R3	M2
33	M2	K1	R3

Table 7.2: Experimental design.

X denotes the news story used (M for Milan plane crash, R for RI fire and K for sinking of the Kursk) and where # denotes which system output was shown to the user (output of system 1, 2 or 3, as described below).

Systems evaluated

In the experiment, users completed a task using the original set of documents for the respective breaking news story as well as the output of one of three IR systems. The three system settings used in the experiment are described below.

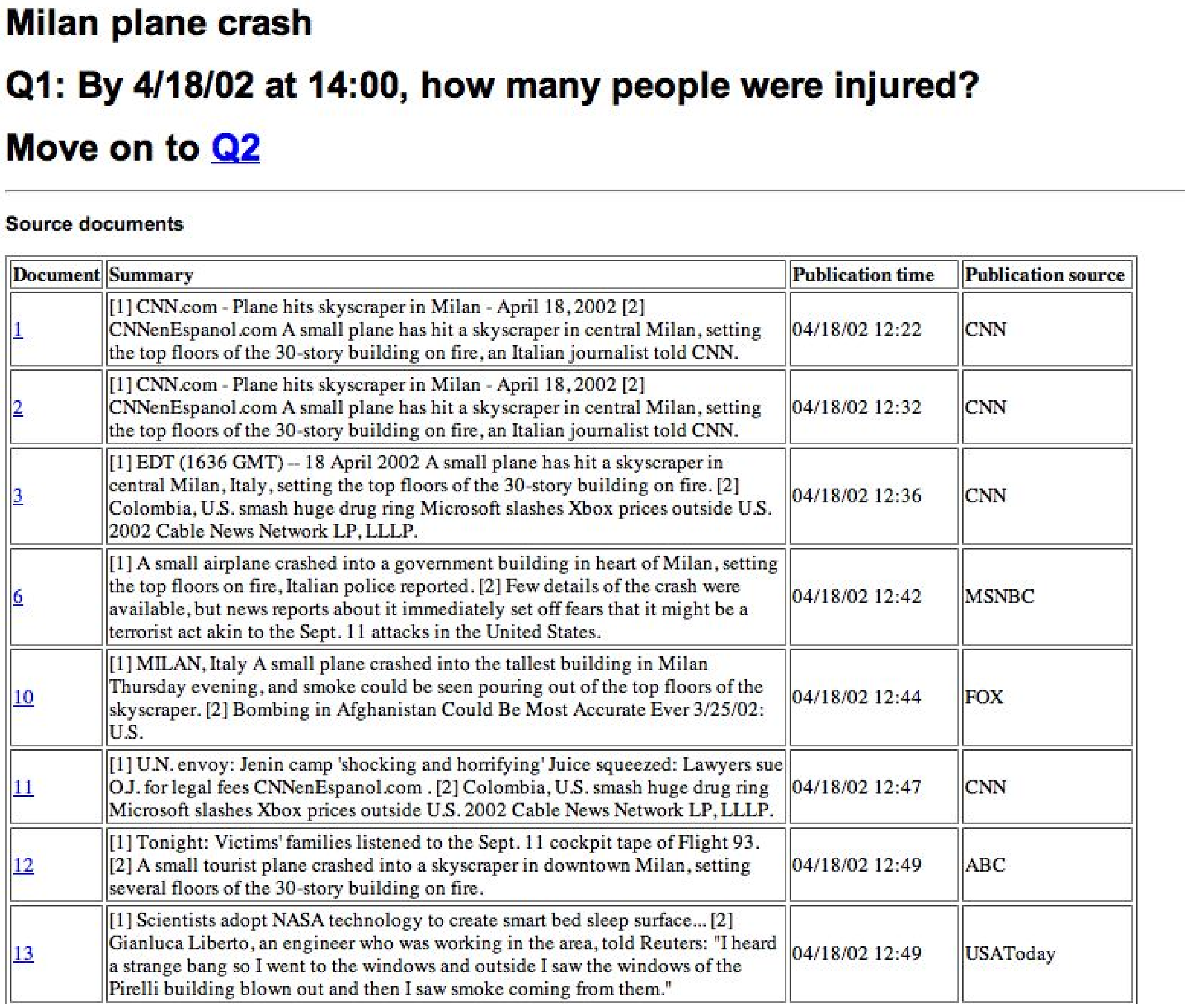
1. The first setting was the baseline system. Users were given a generic (i.e. not question sensitive) summary for each article in the set of documents relevant to the story. In particular, a summary consisting of 2 sentences was produced for each article in the cluster, using the default settings of the MEAD summarizer [101]. This system presents the user with information that is similar to what he or she could obtain from a search engine. For example, given a user's query, the Google search engine[1] returns the ranked list of relevant documents and provides a "snippet," or short summary, for each retrieved document. However, one difference is that here, documents are arranged by publication date and time (earliest to most recent) rather than by relevance to the user's question.

2. The second system produced a question-focused summary using the method described in Chapter VI. In other words, its output was the set of 20 sentences deemed to be most relevant to the input question. The sentences were presented to the user with their respective publication times and sources, and were sorted by relevance to the question. (This setting presented the system's output without the optional answer extraction phase described previously in Section 6.2.)

3: The output of the full system, including the answer extraction option, was shown in the third setting. As described in Chapter 6.2, given the input set of articles about a news story and the question of interest, the output was the top 10 answers found by the system. The answers were arranged by relevance to the question and were shown with their respective publication times and sources.

The same minimal Web-based interface was used to display the different systems' output. Figures 7.1, 7.2 and 7.3 illustrate the system outputs for the first question

[1] $http://www.google.com$



Milan plane crash

Q1: By 4/18/02 at 14:00, how many people were injured?

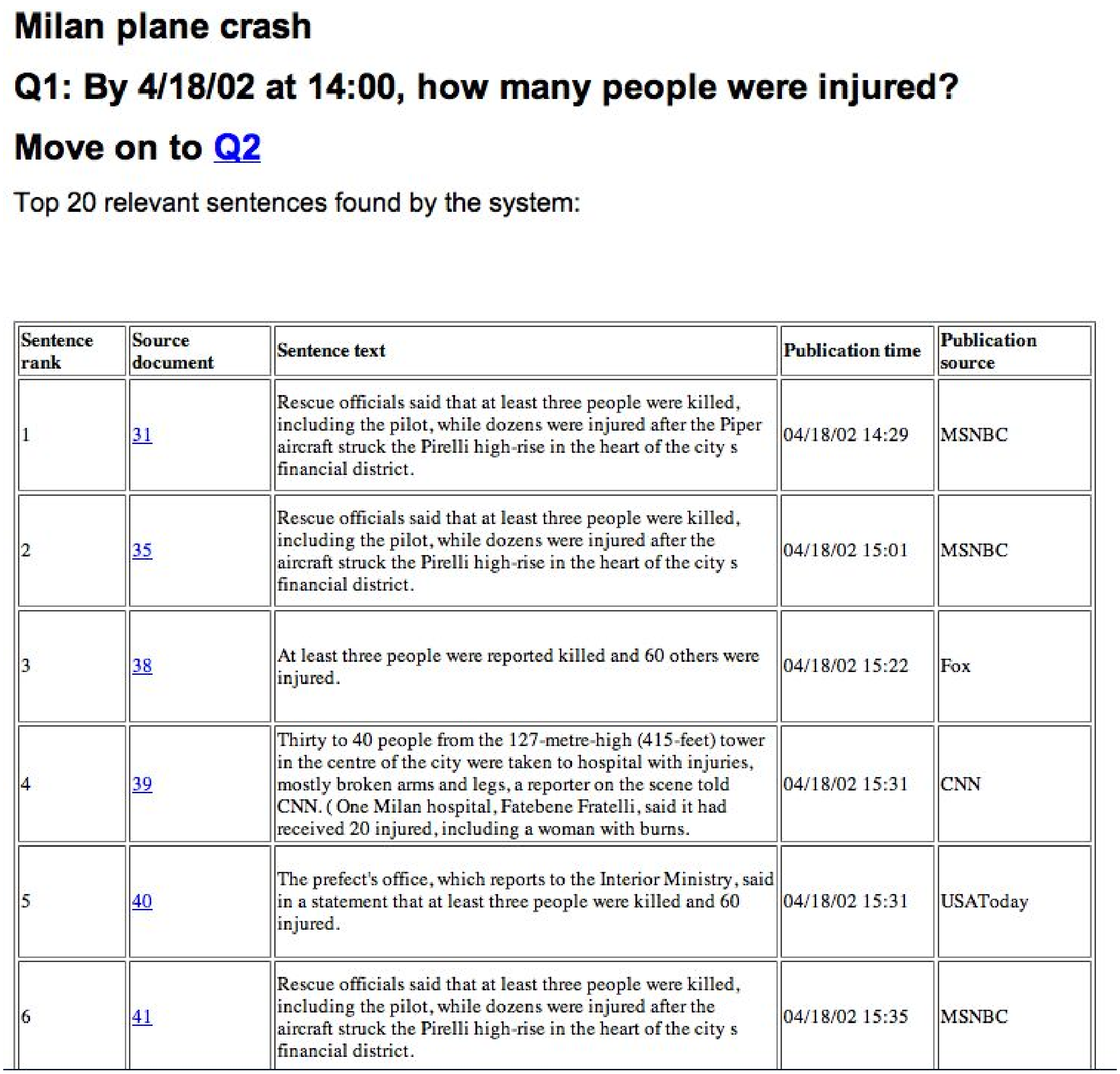
Move on to Q2

Source documents

Document	Summary	Publication time	Publication source
1	[1] CNN.com - Plane hits skyscraper in Milan - April 18, 2002 [2] CNNenEspanol.com A small plane has hit a skyscraper in central Milan, setting the top floors of the 30-story building on fire, an Italian journalist told CNN.	04/18/02 12:22	CNN
2	[1] CNN.com - Plane hits skyscraper in Milan - April 18, 2002 [2] CNNenEspanol.com A small plane has hit a skyscraper in central Milan, setting the top floors of the 30-story building on fire, an Italian journalist told CNN.	04/18/02 12:32	CNN
3	[1] EDT (1636 GMT) -- 18 April 2002 A small plane has hit a skyscraper in central Milan, Italy, setting the top floors of the 30-story building on fire. [2] Colombia, U.S. smash huge drug ring Microsoft slashes Xbox prices outside U.S. 2002 Cable News Network LP, LLLP.	04/18/02 12:36	CNN
6	[1] A small airplane crashed into a government building in heart of Milan, setting the top floors on fire, Italian police reported. [2] Few details of the crash were available, but news reports about it immediately set off fears that it might be a terrorist act akin to the Sept. 11 attacks in the United States.	04/18/02 12:42	MSNBC
10	[1] MILAN, Italy A small plane crashed into the tallest building in Milan Thursday evening, and smoke could be seen pouring out of the top floors of the skyscraper. [2] Bombing in Afghanistan Could Be Most Accurate Ever 3/25/02: U.S.	04/18/02 12:44	FOX
11	[1] U.N. envoy: Jenin camp 'shocking and horrifying' Juice squeezed: Lawyers sue O.J. for legal fees CNNenEspanol.com . [2] Colombia, U.S. smash huge drug ring Microsoft slashes Xbox prices outside U.S. 2002 Cable News Network LP, LLLP.	04/18/02 12:47	CNN
12	[1] Tonight: Victims' families listened to the Sept. 11 cockpit tape of Flight 93. [2] A small tourist plane crashed into a skyscraper in downtown Milan, setting several floors of the 30-story building on fire.	04/18/02 12:49	ABC
13	[1] Scientists adopt NASA technology to create smart bed sleep surface... [2] Gianluca Liberto, an engineer who was working in the area, told Reuters: "I heard a strange bang so I went to the windows and outside I saw the windows of the Pirelli building blown out and then I saw smoke coming from them."	04/18/02 12:49	USAToday

Figure 7.1: Setting 1 output for Milan Q1. Here, the first 13 (of 56) documents in the set are shown, along with a 2-sentence MEAD summary, publication time and source.

related to the Milan plane crash story, as shown to the subjects. Figure 7.4 illustrates the chronological lists of the source documents for the news story, which is shown at the bottom of the page in both settings 2 and 3. In other words, in addition to the system outputs in these settings, the users also have access to the list of all news articles collected, as well as their respective publication time and source.



Milan plane crash

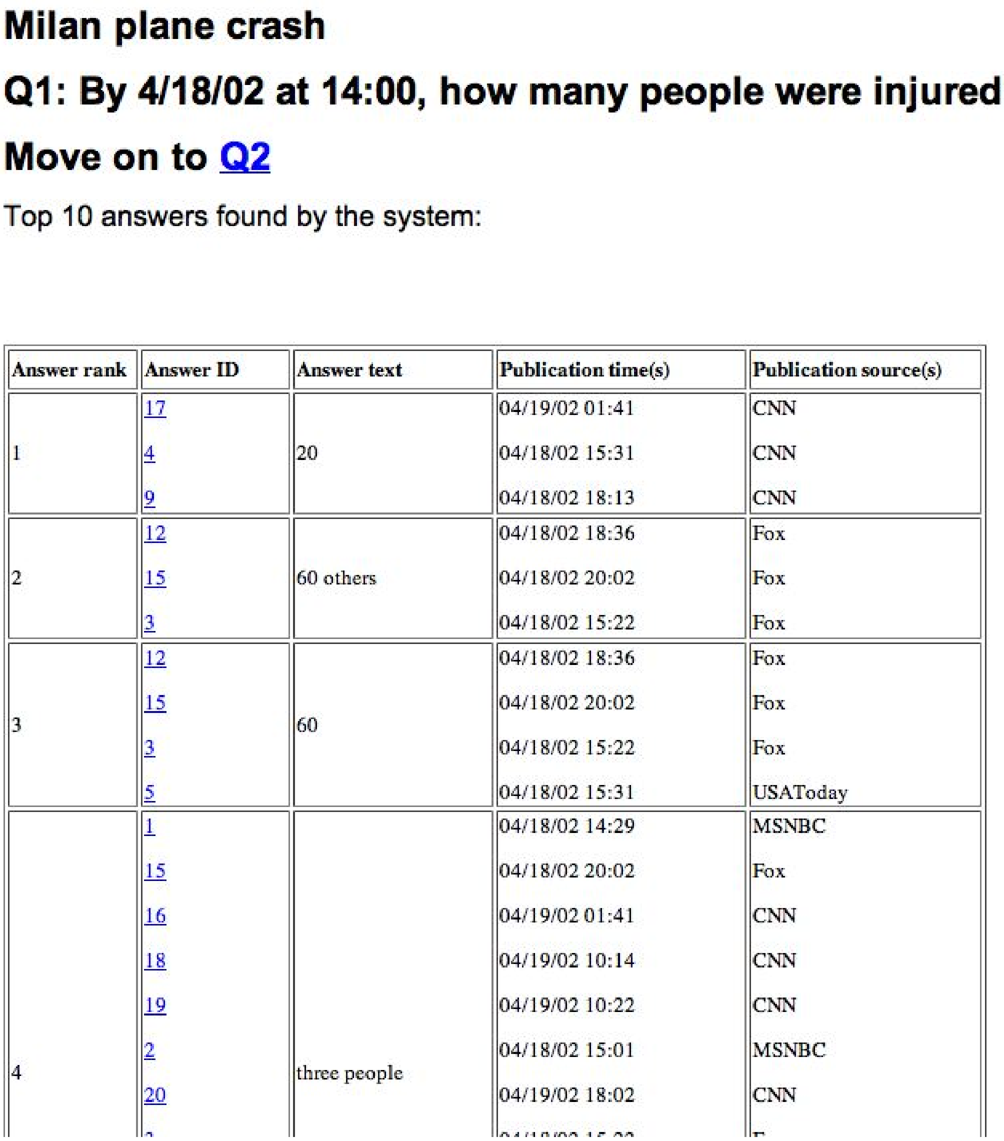
Q1: By 4/18/02 at 14:00, how many people were injured?

Move on to Q2

Top 20 relevant sentences found by the system:

Sentence rank	Source document	Sentence text	Publication time	Publication source
1	31	Rescue officials said that at least three people were killed, including the pilot, while dozens were injured after the Piper aircraft struck the Pirelli high-rise in the heart of the city s financial district.	04/18/02 14:29	MSNBC
2	35	Rescue officials said that at least three people were killed, including the pilot, while dozens were injured after the aircraft struck the Pirelli high-rise in the heart of the city s financial district.	04/18/02 15:01	MSNBC
3	38	At least three people were reported killed and 60 others were injured.	04/18/02 15:22	Fox
4	39	Thirty to 40 people from the 127-metre-high (415-feet) tower in the centre of the city were taken to hospital with injuries, mostly broken arms and legs, a reporter on the scene told CNN. (One Milan hospital, Fatebene Fratelli, said it had received 20 injured, including a woman with burns.	04/18/02 15:31	CNN
5	40	The prefect's office, which reports to the Interior Ministry, said in a statement that at least three people were killed and 60 injured.	04/18/02 15:31	USAToday
6	41	Rescue officials said that at least three people were killed, including the pilot, while dozens were injured after the aircraft struck the Pirelli high-rise in the heart of the city s financial district.	04/18/02 15:35	MSNBC

Figure 7.2: Setting 2 output for Milan Q1, which shows the sentences deemed to be relevant to the question, organized by relevance. Here, the top 6 ranking sentences are shown for Q1.



Milan plane crash

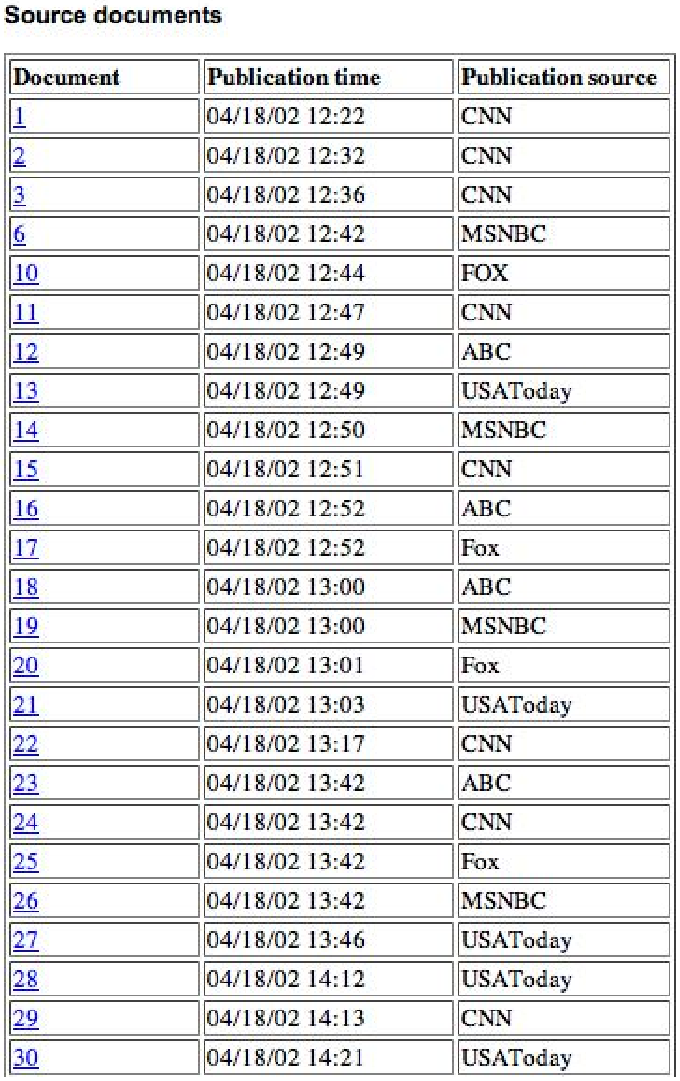
Q1: By 4/18/02 at 14:00, how many people were injured?

Move on to Q2

Top 10 answers found by the system:

Answer rank	Answer ID	Answer text	Publication time(s)	Publication source(s)
1	17 4 9	20	04/19/02 01:41 04/18/02 15:31 04/18/02 18:13	CNN CNN CNN
2	12 15 3	60 others	04/18/02 18:36 04/18/02 20:02 04/18/02 15:22	Fox Fox Fox
3	12 15 3 5	60	04/18/02 18:36 04/18/02 20:02 04/18/02 15:22 04/18/02 15:31	Fox Fox Fox USAToday
4	1 15 16 18 19 2 20	three people	04/18/02 14:29 04/18/02 20:02 04/19/02 01:41 04/19/02 10:14 04/19/02 10:22 04/18/02 15:01 04/19/02 18:02	MSNBC Fox CNN CNN CNN MSNBC CNN

Figure 7.3: Setting 3 output for Milan Q1, which shows the answers extracted by the system for Q1. The answers are ordered by rank (answer of rank 1 is the answer the system deems most likely to be correct). They are shown with publication time and source of the documents reporting the given answer. Here, the top ranking 4 answers for Q1 are shown.



Source documents

Document	Publication time	Publication source
1	04/18/02 12:22	CNN
2	04/18/02 12:32	CNN
3	04/18/02 12:36	CNN
6	04/18/02 12:42	MSNBC
10	04/18/02 12:44	FOX
11	04/18/02 12:47	CNN
12	04/18/02 12:49	ABC
13	04/18/02 12:49	USAToday
14	04/18/02 12:50	MSNBC
15	04/18/02 12:51	CNN
16	04/18/02 12:52	ABC
17	04/18/02 12:52	Fox
18	04/18/02 13:00	ABC
19	04/18/02 13:00	MSNBC
20	04/18/02 13:01	Fox
21	04/18/02 13:03	USAToday
22	04/18/02 13:17	CNN
23	04/18/02 13:42	ABC
24	04/18/02 13:42	CNN
25	04/18/02 13:42	Fox
26	04/18/02 13:42	MSNBC
27	04/18/02 13:46	USAToday
28	04/18/02 14:12	USAToday
29	04/18/02 14:13	CNN
30	04/18/02 14:21	USAToday

Figure 7.4: Source documents for the Milan plane crash story, organized by publication time. This list appears on the bottom of the screen in both systems 2 and 3. Here, the first 30 (of 56) documents in the set are shown.

7.1.3 Instructions and tasks

To give the subjects a context in which to complete the information-seeking tasks, they were asked to imagine that they worked for a government agency responsible for managing emergency situations. They were also told that their individual role in the organization was to monitor the electronic news media for information about emergency events. They were told that they would complete three information-seeking tasks in which they would be asked to answer a set of five questions about a given emergency event. To complete each task, they would be given a set of articles that are relevant to the story, presented in chronological order and with their respective publication time and source information, as well as the output from an information retrieval system. They were not given any information as to what the IR systems did or how the system output was presented. However, they were told that they would be using the output of a different system for each of the three tasks.

The subjects were told that for each information-seeking task, they would be given up to 15 minutes to complete it, but that their goal was to provide "the best, most accurate" answers to the questions as quickly as possible. They were also warned that the correct answer to a question might not remain the same over time, such that they "often need to be sure that the answer provided won't change later on in time or be refuted by another news source." In addition to the printed sheet of instructions, the subjects were shown a sample task consisting of the title, description and document publication time range of a sample news story (the Columbia space shuttle disaster) as well as a set of five questions about the story. They were not asked to actually complete the sample task. Rather, they were asked to read through it in order to verify that they understood what they were being asked to do during the experiment. The instructions and sample task, as shown to the subjects, are

News Story	Questions
Milan plane crash	Q1: By 04/18/02 at 14:00, how many people were injured? Q2: What was the final report on the number of people killed? Q3: What was the plane's origin and destination? Q4: What was the pilot's identity? Q5: Which news source correctly reported the type of plane (make or model) first?
RI nightclub fire	Q1: How many people were in the club at the time of the fire? Q2: When did the reported number of people killed pass 50? Q3: According to CNN, how long did it take for the entire building to burn? Q4: What was the final death toll? Q5: How many people were injured?
Sinking of Kursk submarine	Q1: According to APW, when would the British rescue team arrive? Q2: How was the submarine damaged? Q3: How far down did the Kursk sink? Q4: Consider the question "What caused the Kursk to sink?" At which point in time did a theory emerge that the Kursk might have hit another Russian vessel? Q5: How long could the oxygen supply on board the Kursk last?

Table 7.3: Questions making up the task for each news story.

shown in Appendix A.

Tasks

The current information-seeking task is similar to the information synthesis task, in which a user wants to get answers to a set of questions, with the goal of understanding the basic information surrounding a given topic or story [10]. However, in the current study, users were assigned questions that were deemed as being key to understanding the story and to which the answers were known to vary by time and/or news source. The five questions used for each of the three breaking news stories are shown in Table 7.3.

For a given task, the subjects were given the title of the news story, a brief description of it and the publication time range of the documents in the set. They were asked to indicate how familiar they were with the story, by selecting one of the following options: "I have never heard of this news story," "I know of this story but I do not recall any specific details about it" or "I know of this story well enough to recall some of the main details." They were then asked to read through the set of all five questions before beginning to search for answers to the questions. After finding the answer to a given question, the subjects were asked to indicate their level of agreement with two statements: "I feel confident about my answer" and "It was easy to find the answer to this question." Subjects indicated their level of agreement on a continuous scale from 1 to 5 (on which 1 means "not at all," 3 means "marginally" and 5 means "to a great extent"). These two items were designed to measure the subjects' perceived difficulty in using the IR system's output to find the answer to the question. As mentioned previously, subjects were given up to 15 minutes to complete each task. The time taken to complete each task was recorded in minutes and seconds.

Once a subject had completed all three tasks, he or she had worked with the output of all three systems. Upon completion, the subject was given a brief exit interview, which was designed to elicit feedback about the system designs as well as the subject's overall experience in the information-seeking task. The subjects were given some time to revisit the outputs of the three systems and were asked to answer the questions in writing. Next, the researcher went over the responses with the subject orally and in front of the computer, in order to better clarify the subject's experience. The four questions posed to the subject were:

- **Q1:** Of the three system outputs that you used to complete the task, which one did you like the best? Why?
- **Q2:** Of the three system outputs you used, which one did you like the least? Why?
- **Q3:** What changes would you suggest in improving these systems? Why?

- **Q4:** Would you like to share any other thoughts or comments about your experience?

7.1.4 Variables studied

In evaluating users' effectiveness on a search task, the time to complete a task is often used in information visualization research (e.g. [127], [37]). To contrast, task accuracy is typically used as the primary response variable in evaluating IR systems such as those built in the TREC initiatives (e.g. [61], [119], [87]). In other words, it is assumed that an effective IR system will enable users to be more accurate and to take less time in finding the desired information. Finally, the length of the user's search, measured in terms of the number of documents read while performing a search task, is another measure of retrieval system effectiveness [31]. Here, the assumption is that a good IR system should reduce the number of documents the user must read before finding the desired information. This metric has been used in evaluating Web-based systems including search engines (e.g. [120]).

In the current study, data was collected on six variables, including four objective and two subjective measures. For each task completed by each subject, the following data was collected:

- **Time:** The time (in minutes and seconds) taken to answer the set of five questions. (The maximum time allowed was 15 minutes.)

- **Accuracy:** The proportion of questions correctly answered. (The "gold standard" answers were determined by two annotators who were given unlimited time to find the answers to the questions.)

- **Proportion answered:** The proportion of questions the subject attempted to answer.

- **Source documents accessed:** The number of full-text news articles the subject viewed. This information was obtained from the web log of each session.

- **Perceived confidence:** The mean of the confidence scores assigned by the subject over the five questions in the task. In cases where the subject did not attempt all questions, this was averaged only over the questions that were answered.

- **Perceived ease of task:** The mean of the scores assigned by the subject over the five questions in the task. In cases where the subject did not attempt all questions, this was averaged only over the questions that were answered.

7.2 Analysis of Task Data

In this section, I will compare the three systems with respect to the variables described in Section 7.1.4. In particular, I will examine if the use of system effects the users' task performance. In addition, questions of user effort and confidence in completing the tasks will be examined.

7.2.1 Time to complete task

Although time was originally intended to be a response variable, it was found that it provided very little information about how well each system facilitated the completion of the tasks. As previously mentioned, the subjects were given up to 15 minutes to complete each task (set of 5 questions). However, their instructions were to provide the answers to the questions as quickly and as accurately as possible. These two instructions may be seen as being in conflict with one another, and in fact, among the 99 trials (33 subjects and 3 tasks each), the subjects finished the task early in only 34 cases (34% of the trials). It should be noted that other researches have

System	Mean (minutes)	Median (minutes)
1	13.8	15
2	13.5	15
3	14.1	15

Table 7.4: Mean and median time to complete task, by system.

also noted a tendency for subjects to use all of the time given to them in completing a task when they are asked to be accurate (e.g. [61]).

Based on observations of subjects during the experiment, it is likely that there were more cases where subjects finished the respective task early. However, they may have often used the remaining time to check over their answers rather than to stop early. Therefore, time might be considered to be a measure of task effort rather than performance. Another possibility is that time is correlated to the subjects' skill in searching for information. For example, in comparing the 34 cases where the users finished before the 15 minutes was up, to the other 65 cases where they did not finish early, we observe that the fast searchers were significantly more accurate (mean accuracy of 0.835 versus 0.713, which corresponds to a p-value of 0.0002). They were also more confident in their answers (mean confidence scores of 4.15 versus 3.84 for those who didn't finish early, which corresponds to a p-value of 0.0073).

In summary, over all 99 experimental settings, the time to complete the task followed a slightly skewed distribution (mean of 13.8 minutes; median of 15 minutes). The mean and median time to completion for each test setting are shown in Table 7.4. None of the differences between the systems is statistically significant. I conclude that time is most likely related to user searching skill or effort rather than to system performance.

System	Mean accuracy
1	0.79
2	0.72
3	0.76

Table 7.5: Mean task accuracy, by system.

7.2.2 Task accuracy

Over all 99 trials, task accuracy (measured as the proportion of questions correctly answered by the subject) was normally distributed, with a mean accuracy of 0.76. Table 7.5 shows the mean accuracy per each system. The only difference in mean accuracy between the various settings that is significant is between systems 1 and 2 (mean accuracy of 0.79 versus 0.72). This difference has a p-value of 0.07.

An analysis of variance (ANOVA) was performed in order to see if the system effect on accuracy was still significant, when controlling for measures of the subjects' effort and skill on the tasks. For the first analysis, the response variable was task accuracy and the explanatory variables were the system used (as a categorical variable), whether or not the user attempted all questions in the task (1 if the proportion of questions attempted was 1 and zero otherwise), and whether or not the user finished the task early (1 if the time on task was less than 15 minutes and zero otherwise). Table 7.6 shows the result of this ANOVA, in which the default case (used to compute the constant in the model) is that of a user who used system 3, who completed the task early and who attempted all questions in the given task. It can be seen that when the measures of user skill and effort are included in the model, the system effect is no longer significant. While the coefficient on the effect of system 2 is negative, indicating that when all other factors are controlled, someone using system 2 would on average achieve a task accuracy slightly less than that of the default, the p-value (of 0.319) confirms that this effect is not significant.

Effect	Coefficient	p-value
System 1	0.0207	0.552
System 2	-0.0347	0.319
Time 0	-0.0539	0.096
Attempt 0	-0.2078	0.000
Model constant	0.8376	0.000

Table 7.6: ANOVA results with task accuracy as the response and system as the explanatory variable, controlling for time spent on task and whether or not all questions were attempted.

Effect	Coefficient	p-value
System 1	0.0232	0.561
System 2	-0.0328	0.410
Time 0	-0.1178	0.001
Model constant	0.8371	0.000

Table 7.7: ANOVA results with task accuracy as the response and system as the explanatory variable, controlling for time spent on task.

When time is the only control variable, with system as the only explanatory variable, the system effect is again not significant. As can be seen in Table 7.7, whether or not a user finished the task early is the only significant factor. Therefore, we can conclude that when the effects of effort measures are controlled, there is no significant difference between the three systems as far as task accuracy is concerned.

7.2.3 Number of source documents viewed

The number of source documents that a subject accessed while performing a given task can be viewed as a measure of how hard he or she has to work in searching to find relevant information [31, 120]. In other words, the intuition is that the more helpful a system is for performing the task, the fewer full-text news articles that the user should have to access and read in order to find the answers. The average number of news articles (or source documents) viewed per task is shown in Table 7.8, and is broken out by setting. In system 2, in which the top 20 relevant sentences are shown to the user, the users accessed significantly fewer source documents as compared to the first system, in which they were shown only the list of documents by publication

Setting	**Mean # documents**
1	14.1
2	11.5
3	16.3

Table 7.8: Mean number of source articles viewed, by test setting.

time as well as generic summaries of each article. The p-value associated with this difference is 0.06. In addition, the second setting required the user to access fewer documents than in the third, which showed users the top 10 answers to the question (p-value of 0.0003). However, the differences between the first and third settings were not significantly different with respect to this variable.

One might argue that differences in the number of documents accessed might be the result of other variables that are correlated to the effort and skill of the subjects, such as the number of questions attempted in the time allowed, the time spent on the task, or the subjects' task accuracy. Therefore, an ANOVA was performed in which the number of documents viewed was the response variable and the explanatory variables were the system, whether or not the user finished the task early, whether or not the user achieved 100% accuracy and whether or not the user attempted all five questions in the task. Table 7.9 shows the coefficient for each effect, and its respective p-value. The default experimental setting (used to calculate the model constant) in this case is the use of system 3, for a user who finished the task early, had perfect task accuracy and who attempted all questions.

While the time and the proportion of questions attempted had no significant effects, the effects of the task accuracy and the use of system 2 were both statistically significant (p-values of 0.08 and 0.003, respectively). Therefore, when time spent on task, accuracy and questions attempted are controlled, we still observe a significant system effect on the number of source documents viewed. Therefore, we conclude

Effect	Coefficient	p-value
System 1	-2.522	0.124
System 2	-4.864	0.003
Time 0	1.519	0.329
Accuracy 0	-3.235	0.081
Attempt 0	-2.551	0.156
Model constant	18.577	0.000

Table 7.9: ANOVA results with number of documents viewed as the response and system as the explanatory variable, controlling for accuracy, questions attempted and time spent on task.

Setting	Mean confidence	Mean ease
1	4.06	3.65
2	3.88	3.58
3	3.89	3.43

Table 7.10: Mean perceived confidence and ease of task, by test setting.

that when using system 2 to perform the task, the subjects need to access fewer full-texts articles than they do when using either of the other two systems, in order to achieve the same level of performance in the same amount of time.

7.2.4 Perceived confidence and ease in answering questions

We now turn to the two subjective measures, perceived confidence in answers and the ease of finding answers, as measured on a continuous scale from 1 to 5. Table 7.10 shows the average confidence and ease ratings for each system setting. There is a significant difference between the average confidence rating between settings 1 and 3 (with a p-value of 0.09). However, there are no other significant differences between the test settings on these two measures.

7.3 Analysis of Exit Interview Data

In this section, I analyze the qualitative data that was collected from the users after they had worked with all three of the systems in performing the search tasks. As previously mentioned, this data was collected from all users in an exit interview.

System	Best	Worst
1	18 (0.55)	5 (0.15)
2	13 (0.39)	0 (0)
3	2 (0.06)	28 (0.85)

Table 7.11: Number (and proportion) of users indicating each system as the best or worst system in the exit interview.

7.3.1 User system rankings

One goal of the exit interview was to determine the users' preferences between the three systems for this particular information-seeking task. Table 7.11 shows the number and proportion of subjects who ranked each system as being the best and the worst. It seems clear that the majority of subjects preferred system 1, although many also liked using system 2. Most users (85%) reported that system 3 (NSIR) was the worst of the three. The next few subsections will highlight some of the common reasons the subjects gave for preferring or disliking a particular system's output.

7.3.2 Reasons for preferring system 1

The exit interview written responses (from the subjects) as well as the researcher's notes from the oral portion of the interview were analyzed in order to identify the most common reasons for liking or disliking a particular IR system, in the context of the current task. It should be noted that since subjects sometimes listed more than one reason for liking or disliking a system, that the categories identified here are not mutually exclusive. Therefore, the counts provided as for how many subjects cited a given reason, simply serve as a means to interpret how common this sentiment was among the subjects.

Here are the most commonly cited reasons for preferring system 1 over the other two systems:

- Many users (14/33) liked the fact that the documents were listed in simple chronological order. They liked not having to interpret the meaning of any relevance rankings, and simply used the chronological ordering to determine where the most recent information was.

- 12 of the subjects stated that the single-document summaries given by system 1 were useful for providing the gist of each document, so that they knew which of the full-texts they should view in finding a particular answer.
- Two less common reasons cited were that the summaries "saved searching time" and provided "enough text" in order to answer the question at hand.

7.3.3 Reasons for preferring system 2

- Eight people thought that the output of system 2 provided just enough information (e.g. "the output provided was appropriate in length") as compared to systems 1 and 3.
- Several subjects (6/33) reported that in terms of providing answers to the questions, system 2 was more accurate than the others, because the answer to the question often appeared in the given sentences.
- Two users noted that, as compared to system 3 in which only the answer text was provided, system 2 "provided the answer in context."
- Two subjects stated that they were able to use system 2 in determining which source articles needed to be viewed in full. Similarly, one person simply noted that system 2 "saved time."
- Two subjects felt that the output of system 2 was easy to interpret and understand as compared to that of the other two systems.

- Only one user noted that the relevance rankings of the sentences were helpful in choosing the best answer to a question.

7.3.4 Reasons for liking system 3

As shown in Table 7.11 above, only 2/33 users rated system 3 as being the best of the three systems. Both of these users stated the same reasons for their choice. They thought that system 3, which lists only ranked answers to the questions, saved them time by presenting only the relevant information with no extra text to read.

7.3.5 Reasons for disliking system 1

Only 5 users rated system 1 as being the worst of the three. The remaining 28 users rated system 3 as being the worst system. All 5 of the subjects who disliked system 1 reported that there was too much information that they had to read in this setting. In addition, one person felt that the generic summaries given by this system were not good, in terms of providing the gist of a given source document.

7.3.6 Reasons for disliking system 3

As mentioned, over all, system 3 appeared to be the least popular with the 33 users. All of the problems cited with respect to system 3 fall into one of four categories:

- The accuracy of the answers is not good enough. (16/33 made such comments.)
- Not enough context was provided in order to interpret the system's answers. (5/33)
- It took too much time to interpret the system's answers. (3/33)
- The answers were not listed chronologically. (2/33)

It is obvious that the answers identified by the system were not considered to be accurate enough to be useful for many users. In addition, the second and third category of problems noted seems to indicate that users found it difficult to interpret the output of this system. In some cases, this was because not enough context was provided along with a given candidate answer, for them to understand and use the system's suggestions. Finally, the answers identified by system 3 were listed by rank (i.e. with respect to the probability of being the answer to the question). This was also the case for the output of system 2, which listed the top 20 ranking sentences, in terms of their relevance to the question. One question that was not answered by the results of the current study is whether or not users actually prefer the chronological ordering (as in system 1) over the case where information is ordered first by relevance to the question and then chronologically (as in systems 2 and 3). For example, while many users reported that they preferred system 1 over the other two because the output was organized chronologically, we cannot tell if the users found the relevance rankings of systems 2 and 3 entirely unuseful, or if they even noticed them at all.

7.3.7 Areas for system improvement

The third question of the exit interview asked the subjects to indicate how the three systems could be improved. The responses to this question varied greatly among the users. The majority of the comments concerned how to present or visualize the retrieved information, rather than what information could be retrieved by the systems. For example, the most common suggestion (reported by 9/33 of the subjects) was to allow users to sort the system output by publication time or source, so that the relevant information could be visualized differently depending on the question one is trying to answer. To contrast, four users felt that system output should always be presented chronologically for this task, regardless of the question being answered or the system one is using.

There were many suggested areas for improvement that were generally beyond the scope of the current work, in the sense that they requested a system design significantly different than that being investigated. For example, some users wished that the systems would have provided a “diff” function, that would allow them to automatically compare the difference between two articles. Another suggested given by three subjects was to provide a list of key words appearing in the articles as well as a list of automatically generated synonyms, in order to provide users with the gist of a given news story. (This idea is essentially similar to the design of system 1, which provides a summary overview of each document about a story and is not sensitive to any input question.)

7.3.8 Discussion of exit interview findings

One surprising finding is that the majority of the subjects stated that they preferred system 1, which provides a generic summary of each news article about a given story. It was expected that the users would prefer system 2, which provides the top 20 ranking sentences, deemed as being relevant to the input question. This is because previous work has shown that users perform better on specific information seeking tasks when given query sensitive information as compared to generic (not query sensitive) summaries of documents in a collection (e.g. [37]). Likewise, users perform better on search tasks when systems display full sentences rather than key words only (e.g. [21, 37]). Therefore, the finding that system 3, which displays extracted answers to questions only, was not well-liked by the users was not especially surprising.

In addition, it has been shown that other factors in addition to chronology, such as topical cohesion, should be taken into account when displaying textual information to users. Barzilay and colleagues studied the problem of how to organize information in extractive summaries produced from a set of topically related news articles [13]. They found that simply organizing information chronologically (i.e. by the publication time of the document from which a given sentence was extracted) was not enough. They found that users preferred summaries that were ordered first by topic and then chronologically. In other words, a constraint was introduced such that topically related sentences were adjacent to one another. In the current experiment, information (extracted sentences and answers) relevant to a question was organized first by relevance ranking and then chronologically. Thus, in theory, the passages should be ordered by their similarity to the input question (most to least similar). In future work on the systems, we might explore if clustering (e.g. by the presence or lack of key question words) of the relevant items might help users' understanding of the output.

Finally, it is again worth noting that relevance judgments are somewhat subjective and not well understood by researchers [44, 82]. Therefore, in cases where subjects may have seen the output of systems 2 and 3 for a particular question, and did not agree that the selected sentences (or answers) were relevant to the given question, they may have quickly rejected the use of the system and concluded that system 1, which does not attempt to judge relevance, was the best. However, given that no instructions or tutorials were given to the subjects on how to use the systems' output or what the three outputs consisted of, it may be the case that system 1, which displayed information very similar to that of common search engines such as Google, was simply more familiar to the users.

7.4 Conclusions from User Study

As presented in Section 7.2, no significant differences in task accuracy were observed between the three IR systems used by the subjects. In particular, while a difference in accuracy was initially noted between systems 1 and 2, when subject skill and effort were controlled (e.g. by accounting for the time spent on task and the proportion of questions attempted), the system effect on accuracy was no longer significant. In addition, there were no significant differences between the three systems with respect to the subjective measures of answer confidence and the ease of the task.

To contrast, a significant difference with respect to the number of full-text articles viewed by the users was observed between systems 1 and 2, and systems 2 and 3. In both cases, when users employed system 2, they viewed fewer source articles as compared to the other two settings. In addition, the system effect was still significant, even when the subject skill and effort was controlled (e.g. by incorporating the variables time, accuracy and proportion of questions attempted). This suggests that on average, in order to achieve the same level of task accuracy with the same amount of effort, the users needed to read fewer news articles to complete the task, as compared to the other two systems. In conclusion, the results indicate that the output of system 2 may is more useful than that of the other two systems, since with it the users fall back on viewing source texts less often. As previously mentioned, in addition to facilitating task accuracy and reducing the amount of time that a user needs to spend on an information searching task, another way that retrieval systems can improve the user's experience is to reduce his or her length of search [31]. If in using two IR systems to solve the same task, the user views fewer source documents with one as compared to the other, this could mean that the better system is able to guide the user to the relevant information more directly. Therefore, it may be the case that the users in the current study were able to spend more time reading and understanding information, rather than trying to locate the relevant news articles. However, in future work, it would be useful to collect more rich information about how the users spent their 15 minutes on task, in order to confirm or reject this hypothesis.

The analysis of the exit interview data indicates that it is system 1, rather than 2, that the majority of the subjects preferred. It may be the case that the users felt more comfortable with the output of system 1, since it simply presented a generic two-sentence summary of each source news article. In other words, there were no system rankings that users needed to interpret. Given the few number of people who discussed the system rankings in their exit interviews, it may have been the case that few people understood what they were, or even noticed them. (Recall that the users were not given any overview or tutorial with respect to what the systems were.)

In addition, the summaries in system 1 (with links to the full-text of the articles) were listed in simple chronological order. To contrast, the output of systems 2 and 3 were organized first by relevance (or system ranking) and then by publication date and time. This may have been an unfair disadvantage for these two systems, given the time-sensitive nature of this particular information-seeking task. In future work, it would be interesting to incorporate the popular suggestion put forth by the users, of incorporating a sorting feature (by publication time or source) into systems 2 and 3, and then to reevaluate these systems against the baseline system 1.

In summary, the results of the user study indicate that using the question-focused passage retrieval mechanism, which was incorporated into system 2, helps users in the short-term event tracking task. However, the NSIR answer extraction module, which further processed the output of the passage retrieval component, clearly needs to be improved (with respect to answer accuracy) in order to be of direct use to users in this task. At the same time, another finding was that even generic summaries of individual documents, which are not at all question-sensitive, ordered chronologically, were helpful to the users in seeking answers to the questions in the three tasks. Although the users had to be more proactive in searching the source texts to find answers, their task performance was the same given this particular task and the time allowed to complete it.

In future work, it seems promising to continue to develop the approach taken in system 2, with the question-focused passage retrieval. This approach was shown to be effective at retrieving sentences relevant to an input question (in the evaluation in Chapter VI) and in the user study, obtained promising preliminary results. However, in a future user study, several things could be improved in the current experimental design. One is that more rich data should be collected, in order to examine how users' search strategies might differ when using the system output that is not query sensitive (system 1) versus the output that is query sensitive (system 2). For example, it would be useful to know how much time the users spend on finding the relevant documents (i.e. the documents that contain answers to the questions) and how much time they spend reading and understanding the documents. Also, the users could be asked to describe the full evolution of an answer to a time or source-sensitive question (rather than simply to find the best answer). Finally, another limitation of the current study was that the users were given the output of pre-assigned questions. In other words, they did not use the system to find answers to their own questions about the stories, because of the need to ensure that the task involved time and source-sensitive questions. One possible design for a future study, would be to allow users to use the system for a limited amount of time, asking any questions of their choosing, and then to ask them to write a report about the story and to include a timeline of events surrounding it. In conclusion, the current study presented promising preliminary results for system 2. In future work, I hope to both improve the system itself based on the findings in my initial study. In addition, I plan to conduct a more extensive user study in order to further examine how this system differs from the baseline, when users employ it to learn about time-sensitive stories as told from multiple news sources.

CHAPTER VIII

Discussion and Conclusion

The central hypothesis of the current thesis is that users can better follow the events described in dynamic, online news stories using an information retrieval system that is specifically designed for this task as compared to using existing systems that do not take into account the time and source sensitivity of this genre of textual information. Another hypothesis is that an effective, initial version of such a system can be built for this purpose, using existing tools. In Chapter I, the introduction, it was argued that current IR systems, such as text summarizers and question answering systems, do not support users attempting to follow dynamic information since they do not take the relationships between time, source and reliability into consideration when presenting information to users. Therefore, a prototype system, which was based on the MEAD summarizer, was designed, implemented and evaluated.

8.1 The Challenges of Dynamic News Texts

One of the goals of the thesis was to build a corpus of breaking news stories and to examine how they conveyed change over time and across different sources. To this end, the first three studies examined different aspects of information change in the stories. Chapter III describes the first study, which involved building a corpus of breaking news stories and undertaking an empirical analysis in which change over time was examined at the sentence level. To contrast, Chapter IV details experiments undertaken at the document level, which attempt to model changing information in a news story as an evolutionary process. Finally, the third study, an annotation experiment described in Chapter V, examined the extent to which users agreed on the sentences containing answers to factual questions over time, and which answers provided new, previously unseen information.

8.1.1 Empirical study

In Chapter III, a corpus of breaking news stories was created that was manually annotated for key factual questions, as well as the sentences providing answers to the questions. A corpus analysis highlighted some of the challenges that this genre of dynamic text presents to IR systems. First of all, it is clear that anyone wishing to use online news to follow events of public interest over time needs to follow news from multiple sources. This is because sources often contradict one another as to what the current facts in a story are, especially in the early stages of an investigation. At the same time, the reported facts are likely to change over time, until the point when all sources report the same (or similar) information. Out of all the answers manually identified in the corpus, less than 15% represented settled information, that did not change later in time.

In analyzing sets of extracted answers to a given question, it was hypothesized that there would be a negative relationship between the lexical similarity of answer pairs and their publication time span. However, in this corpus, little evidence was found to support this claim, even though a number of similarity metrics were examined. To contrast, lexical similarity was correlated to source. In other words, lexically similar answers to a question are more likely to be from the same news source, as compared to two answers that are not very similar. In summary, the conclusion of the empirical analysis was that complex relationships, such as contradiction, information subsumption and paraphrasing, often exist between pairs of answers to questions reported over time and across sources in breaking news stories. It is unlikely that we will be able to model the chronological relationships between reported facts, based on simple similarity metrics.

8.1.2 Recovering chronological relationships between texts

Chapter IV considered the modeling of chronological relationships between news documents rather than individual facts. In particular, a biologically-inspired evolutionary model was proposed, which assumes that all news articles published about a given story evolve over time from a common ancestor. The articles “mutate” over time, with new words appearing, giving rise to many different document “species.” To create the model for a set of articles about a story, the edit distance between each pair of articles was first calculated. Then, given these distances, the evolutionary tree is produced, which shows the most likely evolutionary history of the set of articles. In the reported experiments, the trees were evaluated by the extent to which the chronological relationships could be recovered by simply transversing the tree.

The experimental results showed that the technique could be used to recover chronological relationships between sets of documents that were published within short time spans of one another. For sets of news articles published within longer time periods (e.g. over the course of a few months or a year), the performance was not as good. This seems to suggest that some news stories have a more evolutionary nature than others. For example, some stories detail changing information quickly or over a short period of time, whereas other stories evolve at a slower pace. In some sense, it is promising that the chronology recovery technique was more useful in the case of sets of articles that were published within shorter time periods of one another.

This is because it is for this type of data that users are more likely to need IR tools to track information, since they need to be able to find and read information quickly as it is reported.

In addition, another finding is that it is easier to automatically detect change at the document level, rather than at smaller levels of granularity. For example, in Chapter III, no significant correlations were found between pairs of sentences containing answers to the same question, and their chronological differences for a number of lexically-based similarity metrics. One reason that it may be easier to detect information change between two news documents is that the order in which information is presented is likely to signal temporal clues. In fact, in updating a previously introduced news story over time, journalists often present new information first in an article, and may move older information to the later parts of the article [81]. Therefore, analyzing stories at the sentence level would not pick up on such temporal clues, if the sentence itself was not changed from one publication time to the next.

8.1.3 Identifying novel information over time

Finally, another key challenge in processing time-sensitive news articles over time is the issue of what constitutes new information. As previously mentioned, in Chapter III, subtle yet non-trivial relationships exist between sentences that are relevant to the same question. This fact was also clearly evidenced in the annotation experiment described in Chapter V. In this experiment, annotators were first asked to identify the set of sentences that contain answers to a given question. Next, they were to review their set of relevant sentences (that contain answers to the question) in chronological order, and to eliminate those sentences that did not contain a new, previously unseen answer to the question. In other words, this second task was to identify the set of novel answers to a question.

While the annotators agreed on which sentences contained answers to questions, the agreement on which sentences were novel was extremely low. This result suggests that what constitutes “novel information” that becomes available over time, is not a very objective question. Therefore, although novelty identification at the sentence level has recently been an active area of IR research, for the problem of finding answers to specific factual questions, it is more useful to focus on the passage retrieval task, from a set of articles published over time and by various sources. In addition,

8.2 A System to Support Short-term Event Tracking

Given the challenges of processing time-sensitive news articles, and the fact that, in general, users do not agree when new factual information has become available over time, a system was designed that focused on retrieving the sentences in a set of breaking news documents that provide an answer to an input question of interest. Rather than performing novelty detection or trying to provide an exact answer to the user’s question, the system returns the top ranking (in terms of relevance to the question) 20 sentences, along with publication source and time information. Although the system is a prototype and can certainly be improved in future work, it fared quite well in the user study detailed in Chapter VII. In particular, it was compared against a baseline, which was designed to display information similar to a readily available search engine such as Google. The baseline system presented a generic 2-sentence “snippet” for each news article to the user, along with the publication time and date. As compared to the baseline, the new system enabled users to view fewer full-text articles, when performing a task involving answering sets of time and source-sensitive questions about a news story. In other words, users were able to obtain similar task performance in the same amount of time, while having to read fewer source articles.

In future work, the system could be improved in a number of ways. For example, once the system is implemented into an interactive, Web-based application, it could allow for users to organize the displayed relevant information in various ways. In the exit interviews conducted at the end of the user study, many subjects mentioned that they would have liked to have had the option of sorting by publication time and source, in addition to relevance ranking, depending on the question that they were trying to answer. Another insight gained from the interviews was that many subjects did not understand the meaning of the relevance rankings or, in some cases, may not have paid any attention to them. Therefore, designing an interface that will enable the users to take full advantage of the system's output will be important in improving its usefulness in supporting the task of following of changing information.

Another area for future work is novelty detection below the sentence level. In the current work, I tested the hypothesis that novelty judgments at the sentence level would be more reliable if they were fact-focused rather than topic-focused. Since many IR systems operate at the sentence level (e.g. text summarization and question-answering systems), it would be beneficial to develop a definition of novelty that could be applied at the sentence level. Unfortunately, as shown in Chapter V, users do not agree on sentence level novelty judgments in the fact-focused or the topic-focused setting. However, previous research has suggested that human subjects do agree on the task of identifying facts in text (e.g. [124, 84]). Therefore, future work should explore the possibility of identifying novel textual units smaller than the sentence.

In conclusion, this thesis undertook the task of developing a prototype IR system that supports the following of time and source-sensitive information in breaking news stories. As discussed in Chapter I, there is a need for systems that are both query-sensitive and time and source-sensitive. However, as the current work has illustrated, there are many challenges to overcome in building such a system. Perhaps the biggest challenge was the fact that there is little consensus between users as to when an answer to a factual question changes. Therefore, the system that was designed did not address the novelty detection problem. Rather, when answering a given factual question, users saw the 20 most relevant sentences, their publication times and sources, and decided for themselves which answer was the most up-to-date and accurate. The results of the user study indicate that even though the system does not fully automate the process of finding the most recent and reliable answer, the users are able to use it effectively in finding the answers to time and source sensitive questions.

APPENDIX

APPENDIX A

Instructions for User Study Participants

In this study, you are to imagine that you work for a government agency that is responsible for managing emergency situations. Your particular role is to monitor the electronic (online) news media for information about such events. For each of three emergency events, you will be given a set of news articles related to the event, published by various news sources at different points in time. They will be given to you in chronological order. You will be given a brief description of a story of interest as well as a set of 5 questions to answer about the story. **Your goal is to provide the best, most accurate answer to the questions as fast as you can.**

This is not always as easy as it seems! In many cases, the "correct" answer to the question may change with time. In addition, different news sources may disagree with one another on what the facts are. Therefore, you often need to be sure that the answer you provide won't change later on in time or be refuted by another news source.

You will be using a different information retrieval system to answer the questions about each news story. Please use only the information expressed in the documents and the information given to you in the system output. In other words, do not use your own world knowledge to answer the questions.

For each question you answer about a story, you will also be asked to indicate your agreement with respect to the following statements: “I feel confident about my answer” and “It was easy to find the answer to this question.” You will be asked to indicate your level of agreement with the statements on a continuous scale from 1 to 5 (where 1 = “Not at all,” 3 = “Marginally” and 5 = “To a great extent.”)

Finally, once you have completed the task for all three news stories, you will be asked a few questions about your experience.

General Instructions for Participants, con't.

Example task

User ID:

Setting:

Time:

Story title: Columbia space shuttle disaster

Story description: Documents in this set describe the NASA Columbia space shuttle disaster that occurred in February of 2003.

Document time range: All news articles were published between 2/1/03 at 07:30 EST and 2/3/03 at 23:00 EST.

How familiar you are with this story (please circle one)?

1. I have never heard of this story.
2. I know of this story but I do not recall any specific details about it.
3. I know this story well enough to recall some of the main details.

Instructions: Please read over all five questions, and then begin.

Questions:

1) By 12:00 on 2/1/03, how many people were confirmed dead?

Please indicate the extent to which you agree with the following questions:

1. I feel confident about my answer.

```
|_______________|________________|_______________|_______________|
1               2                3               4               5
Not at all                   Marginally                 To a great extent
```

2. It was easy to find the answer to this question.

```
|_______________|________________|_______________|_______________|
1               2                3               4               5
Not at all                   Marginally                 To a great extent
```

2) What caused the disaster?

Please indicate the extent to which you agree with the following questions:

1. I feel confident about my answer.

```
|______________|______________|______________|______________|

1              2              3              4              5

Not at all                 Marginally                 To a great extent
```

2. It was easy to find the answer to this question.

```
|______________|______________|______________|______________|

1              2              3              4              5

Not at all                 Marginally                 To a great extent
```

3) According to Fox News, what was the time of the last contact with the shuttle?

Please indicate the extent to which you agree with the following questions:

1. I feel confident about my answer.

```
|_______________|_______________|_______________|_______________|

1               2               3               4                5

Not at all                   Marginally                   To a great extent
```

2. It was easy to find the answer to this question.

```
|_______________|_______________|_______________|_______________|

1               2               3               4                5

Not at all                   Marginally                   To a great extent
```

4) Which parts of the shuttle were found on 2/3/03?

Please indicate the extent to which you agree with the following questions:

1. I feel confident about my answer.

```
|_______________|_______________|_______________|_______________|
1               2               3               4               5
Not at all                   Marginally                    To a great extent
```

2. It was easy to find the answer to this question.

```
|_______________|_______________|_______________|_______________|
1               2               3               4               5
Not at all                   Marginally                    To a great extent
```

5) Where did the shuttle disintegrate?

Please indicate the extent to which you agree with the following questions:

1. I feel confident about my answer.

```
|______________|______________|______________|______________|

1              2              3              4              5

Not at all                 Marginally                 To a great extent
```

2. It was easy to find the answer to this question.

```
|______________|______________|______________|______________|

1              2              3              4              5

Not at all                 Marginally                 To a great extent
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Steven Abney, Michael Collins, and Amit Singhal. Answer Extraction. In *Conference on Applied Natural Language Processing (ANLP)*, 2000.

[2] Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning Search Engine Specific Query Transformations for Question Answerings. In *The 10th World Wide Web Conference (WWW 2001)*, Hong Kong, 2001.

[3] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. In *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, February 1998.

[4] James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of news topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001.

[5] James Allan, Rahul Gupta, and Vikas Khandelwal. Topic models for summarizing novelty. In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, 2001.

[6] James Allan, Hubert Jin, Martin Rajman, Charles Wayne, Daniel Gildea, Victor Lavrenko, Rose Hoberman, and David Caputo. Topic-based novelty detection 1999 summer workshop at clsp final report, August 1999.

[7] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and Novelty Detection at the Sentence Level. In *Association for Computing Machinery (ACM) Conference of the Special Interest Group in Information Retrieval (SIGIR '03)*, Toronto, Canada, 2003.

[8] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321. ACM Press, 2003.

[9] Lisa Allen, Chris Charron, and Sadaf Roshan. Re-engineering the News Business. In *Technical report, The Forrester Group*, June 2002.

[10] Enrique Amigo, Julio Gonzalo, Victor Peinado, Anselmo Peñas, and Felisa Verdejo. An Empirical Study of Information Synthesis Task. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 207–214, Barcelona, Spain, July 2004.

[11] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. Searching the Web. *ACM Transactions on Internet Technology*, 1(1), August 2001.

[12] Thomas Ball and Fred Douglis. An Internet Difference Engine and Its Applications. In *Proceedings of the IEEE COMPCON'96*, February 1996.

[13] Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. Inferring Strategies for Sentence Ordering in Multidocument Summarization. 17:35–55, 2002.

[14] Regina Barzilay and Lillian Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proceedings of NAACL-HLT03*, Edmonton, 2003.

[15] Regina Barzilay and Kathleen McKeown. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL/EACL 2001*, Toulouse, 2001.

[16] Nicholas J. Belkin and W. Bruce Croft. Information Filtering and Information Retrieval. *Communications of the Association for Computing Machinery*, 35(12), 1992.

[17] Charles H. Bennett, Ming Li, and Bin Ma. Chain Letters and Evolutionary Histories. *Scientific American*, pages 76–81, June 2003.

[18] Krishna Bharat and Andrei Broder. A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. In *7th International World Wide Web Conference*, 1998.

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[20] Michael Buckland. What is a Document? *Journal of the American Society of Information Science*, 48(9):804–809, September 1997.

[21] Orkut Buyukkokten, Oliver Kaljuvee, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Efficient Web Browsing on Handheld Devices Using Page and Form Summarization. *ACM Transactions on Information Systems (TOIS)*, 20(1):82–115, January 2002.

[22] Nicola Cancedda. Text generation from MUC templates. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, Toulouse, June 1999.

[23] Jaime G. Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. pages 335–336, Melbourne, Australia, 1998.

[24] Jean Carletta. Assessing Agreement on Classification Tasks: The Kappa Statistic. 22(2):249–254, 1996.

[25] Ying-Ju Chen and Hsin-Hsi Chen. NLP and IR Approaches to Monolingual and Multilingual Link Detection. In *Proceedings of COLING '02*, 2002.

[26] Junghoo Cho and Hector Garcia-Molina. Estimating Frequency of Change. *ACM Transactions on Internet Technology*, 3(3), August 2003.

[27] Charles L. A. Clarke, Gordon V. Cormack, D. I. E. Kisman, and Thomas R. Lynam. Question Answering by Passage Selection (Multitext Experiments for TREC 9). In *NIST Special Publication 500-249: The Ninths Text REtrieval Conference (TREC 9)*, pages 673–683, 2000.

[28] Philip R. Clarkson and Ronald Rosenfeld. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *ESCA Eurospeech*, 1997.

[29] Paul Clough. Measuring Text Reuse and Document Derivation. Technical report, Department of Computer Science, University of Sheffield, February 2001.

[30] Paul Clough, Robert Gaizauskas, Scott S.L. Piao, and Yorick Wilks. Measuring text reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 152–159, 2002.

[31] William S. Cooper. Expected Search Length: A Single Measure of Retrieval Effectiveness Based on Weak Ordering Action of Retrieval Systems. *Journal of the American Society of Information Science*, 19:30–41, 1968.

[32] Jim Cowie and Yorick Wilks. Information extraction. In *Handbook of Natural Language Processing*, pages 241–260. Marcel Dekker, Inc., 2000.

[33] Blaise Cronin, Herbert W. Snyder, Howard Rosenbaum, Anna Martinson, and Ewa Callahan. Invoked on the Web. *Journal of the American Society for Information Science*, 49(14):1319–1328, 1998.

[34] C. A. Cuadra and R. V. Katter. Opening the Black Box of Relevance. *Journal of Documentation*, 23(4):291–303, 1967.

[35] Hang Cui, Min-Yen Kan, and Tat-Sent Chua. Unsupervised Learning of Soft Patterns for Generating Definitions for Online News. In *World Wide Web Conference (WWW 2004)*, New York, New York, May 2004.

[36] Fred Douglis, Thomas Ball, Yih-Farn Chen, and Eleftherios Koutsofios. The AT&T Internet Difference Engine: Tracking and Viewing Changes on the Web. *World Wide Web*, 1(1), 1998.

[37] Offer Drori. How to Display Search Results in Digital Libraries - User Study. In *Proceedings of New Developments in Digital Libraries (NDDL '03)*, Angers, France, 2003.

[38] Gunes Erkan and Dragomir Radev. LexRank: Graph-based Lexical Centrality as Salience in Text. *JAIR*, 22:457–479, 2004.

[39] Jacqui Ewart. Prudence not Prurience: A Framework for Journalists Reporting Disasters. In *Proceedings of the 23rd Annual Conference of the Australia and New ZealandCommunication Association (ANZCA 2002)*, Queensland, 2002.

[40] Joseph Felsenstein. PHYLIP: Phylogeny Inference Package. Technical report, Department of Genome Sciences, University of Washington, 1995.

[41] Elena Filatova and Eduard Hovy. Assigning Time-stamps to Event-clauses. In *Proceedings of ACL Workshop on Temporal and Spatial Information Processing*, Toulouse, France, July 2001.

[42] John G. Fiscus and George R. Doddington. Topic Detection and Tracking Evaluation Overview. pages 17–32, 2002.

[43] Walter M. Fitch and Emanuel Margoliash. Construction of Phylogenetic Trees. *Science*, 155(3760):279–284, January 1967.

[44] Thomas J. Froehlich. Relevance Reconsidered - Towards an Agenda for the 21st Century: Introduction to Special Topic Issue on Relevance Research. *Journal of the American Society for Information Science*, 45(3):124–133, April 1994.

[45] Robert Gaizauskas, Mark Hepple, and Mark Greenwood. Information Retrieval for Question Answering: a SIGIR 2004 Workshop. In *SIGIR 2004 Workshop on Information Retrieval for Question Answering*, 2004.

[46] Eric J. Glover, Gary W. Flake, Steve Lawrence, William P. Birmingham, Andries Kruger, C. Lee Giles, and David M. Pennock. Improving Category Specific Web Search by Learning Query Modifications. In *The Symposium on Applications and the Internet (SAINT 2001)*, San Diego, California, 2001.

[47] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd ACM Conference Research and Development in Information Retrieval*, Berkeley, CA, 1999.

[48] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girji, Vasile Rus, and Paul Morarescu. Falcon: Boosting Knowledge for Answer Engines. In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, pages 479–488, 2000.

[49] Donna Harman. Overview of the TREC 2002 novelty track, 2002.

[50] Geoffrey Harris and David Spark. *Practical Newspaper Reporting.* Butterworth-Heinemann, 1997.

[51] Stephen P. Harter. Variations in Relevance Assessments and the Measurement of Retrieval Effectiveness. *Journal of the American Society for Information Science*, 47(1):37–49, 1996.

[52] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. Simfinder: A flexible clustering tool for summarization. In *NAACL Workshop on Text Summarization*, 2001.

[53] Timothy C. Hoad and Justin Zobel. Methods for Identifying Versioned and Plagiarized Documents. *Journal of the American Society for Information Science and Technology*, 54(3):203–215, 2003.

[54] Gord Hotchkiss. How September 11 Redefined the Web. http://www.searchengineposition.com/info/netprofit/september11.asp, September 2001.

[55] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question Answering in Webclopedia. In *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, pages 655–664, Gaithersburg, MD, 2000.

[56] Hongyan Jing and Kathleen R. McKeown. Cut and Paste-Based Text Summarization. pages 178–185, Seattle, WA, April 2000.

[57] Boris Katz. From Sentence Processing to Information Access on the World Wide Web. In *Natural Language Processing for the World Wide Web: Papers from the 1997 AAAI Spring Symposium*, pages 77–94, 1997.

[58] Christopher M. Kelley and Gillian DeMoulin. The Web Cannibalizes Media. In *Technical report, The Forrester Group*, May 2002.

[59] Scott Kirsner. The Breaking News Dilema. In *Columbia Journalism Review*, November/December 1997.

[60] Wallace Koehler. Web Page Change and Persistence - A Four-Year Longitudinal Study. *Journal of the American Society for Information Science and Technology*, 53(2):162 – 171, 2002.

[61] Jurgen Koenemann and Nicholas J. Belkin. A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96)*, Vancouver, Canada, 1996.

[62] Klaus Krippendorff. *Content Analysis: An Introduction to its Methodology.* Sage Publications, Beverly Hills, CA, 1980.

[63] Julian Kupiec. MURAX: A Robust Linguistic Approach for Question Answering Using an On-line Encyclopedia. In *The 16th SIGIR Conference*, Pittsburgh, PA, 2001.

[64] Oren Kurland and Lillian Lee. Corpus Structure, Language Models, and Ad Hoc Information Retrieval. In *SIGIR 2004*, 2004.

[65] Oren Kurland and Lillian Lee. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *SIGIR 2005*, Salvador, Brazil, August 2005.

[66] C. Kwok, O. Etzioni, and D. S. Weld. Scaling Question Answering to the Web. In *The 10th World Wide Web Conference (WWW 2001)*, Hong Kong, 2001.

[67] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. 33:159–174, 1977.

[68] Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade. UMass at TREC 2002: Cross Language and Novelty Tracks. In *Eleventh Text REtrieval Conference (TREC 2002)*, 2002.

[69] Steve Lawrence and C. Lee Giles. Accessibility of Information on the Web. *Nature*, pages 107–109, 1999.

[70] Raija Lehtokangas and Kalervo Järvelin. Consistency of textual expression in newspaper articles: An argument for semantically based query expansion. *Journal of Documentation*, 57(4):535–548, 2001.

[71] David M. Levy. Fixed or Fluid? Document Stability and New Media. In *ACM European Conference on Hypermedia Technology*, 1994.

[72] Dekang Lin and Patrick Pantel. DIRT: discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, pages 323–328, 2001.

[73] Peter Lyman and Hal R. Varian. How much information 2003, 2003.

[74] Inderjeet Mani, Barry Schiffman, and Jianping Zhang. Inferring Temporal Ordering of Events in News. In *Proceedings of HLT-NAACL 2003 Short Papers*, Edmonton, 2003.

[75] Inderjeet Mani and George Wilson. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong, 2000.

[76] William Mann and Sandra Thompson. Rhetorical Structure Theory: Towards a Functional Theory of Text Organization. *Text*, 8(3):243–281, 1988.

[77] Daniel Marcu. The Rhetorical Parsing of Unrestricted Text: A Surface-Based Approach. *Computational Linguistics*, 26(3):395–448, 2000.

[78] Daniel Marcu and Abdessamad Echihabi. An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 2002.

[79] David Masterson and Nicholas Kushmerick. Information Extraction from Multi-Document Threads. In *Proc. Workshop on Adaptive Text Extraction and Mining*, pages 34–41, 2003.

[80] George A. Miller. Wordnet: A Lexical Database for English. In *Communications of the ACM*, volume 11, pages 39–41, 1995.

[81] Catherine C. Mitchell and Mark D. West. *The News Formula: A Concise Guide to News Writing and Reporting.* St. Martin's Press, New York, 1996.

[82] Stefano Mizzaro. Relevance: The whole history. *Journal of the American Society of Information Science*, 48(9):810–832, 1997.

[83] Tatsunori Mori, Masanori Nozawa, and Yoshiaki Asada. Multi-Answer-Focused Multi-Document Summarization Using a Question-Answering Engine. In *International Conference on Computational Linguistics (COLING 2004)*, Geneva, August 2004.

[84] Ani Nenkova and Rebecca Passonneau. Evaluating Context Selection in Summarization: the Pyramid Method. In *NAACL-HLT '04*, 2004.

[85] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's News on the Web? The Evolution of the Web from a Search Engine Perspective. In *World-Wide Web Conference (WWW 04)*, May 2004.

[86] Jahna Otterbacher, Gunes Erkan, and Dragomir Radev. Using Random Walks for Question-focused Sentence Retrieval. In *Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT-EMNLP '05)*, Vancouver, October 2005.

[87] Paul Over. TREC 6 Interactive Report. In *Proceedings of the Sixth Text Retrieval Conference (TREC 6)*, Maryland, 1998.

[88] Paul Over and James Yen. An Introduction to DUC 2003: Intrinsic Evaluation of Generic News Text Summarization Systems. In *Document Understanding Conference 2003*, 2003.

[89] Larry Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford University, Stanford, CA*, 1998.

[90] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *Proceedings of HLT/NAACL 2003*, Edmonton, Canada, 2003.

[91] Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Association for Computational Linguistics*, 2004.

[92] Craig Partridge, Paul Barford, David D. Clark, Sean Donelan, Vern Paxson, Jennifer Rexford, and Mary K. Vernon. The Internet Under Crisis Conditions: Learning from September 11. http://www.nap.edu/html/internetcrises/internetcrisis.pdf, November 2002.

[93] Jay M. Ponte and W. Bruce Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR 1998*, 1998.

[94] John Prager, Dragomir Radev, Eric Brown, and Anni Coden. The Use of Predictive Annotation for Question Answering in TREC 8. In *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*, pages 399–411, Gaithersburg, MD, 1999.

[95] James Pustejovsky, Jose Castano, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir Radev. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of AAAI Spring Symposium on Question Answering*, Stanford, CA, 2003.

[96] Dragomir Radev. Weakly Supervised Graph-based Methods for Classification. Technical Report CST-TR-500-04, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan, USA, 2004.

[97] Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. Probabilistic Question Answering on the Web. In *World Wide Web Conference (WWW '02)*, Honolulu, Hawaii, 2002.

[98] Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. Probabilistic Question Answering on the Web. *Journal of the American Society for Information Science and Technology*, 56(3), March 2005.

[99] Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. Probabilistic Question Answering on the Web. *Journal of the American Society for Information Science and Technology (JASIST)*, March 2005.

[100] Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha Blair-Goldensohn. NewsInEssence: Summarizing Online News Topics. *Communications of the Association for Computing Machinery (CACM)*, 48(10), October 2005.

[101] Dragomir Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Celebi, Danyu Liu, and Elliott Drabek. Evaluation Challenges in Large-scale Multi-document Summarization: The MEAD Project. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, July 2003.

[102] Dragomir R. Radev. Rendezvous: A WWW Synchronization System. In *Second International WWW Conference Poster Session*, October 1994.

[103] Dragomir R. Radev. A Common Theory of Information Fusion from Multiple Text Sources, Step One: Cross-document Structure. In *Proceedings of the 1st Workshop on Discourse and Dialogue of the Association for Computational Linguistics*, Hong Kong, October 2000.

[104] Dragomir R Radev, Kelsey Libner, and Weiguo Fan. Getting Answers to Natural Language Queries on the Web. *Journal of the American Society for Information Science and Technology (JASIST)*, 2001.

[105] Dragomir R. Radev and Kathleen R. McKeown. Generating Natural Language Summaries from Multiple On-Line Sources. *Computational Linguistics*, 4:469–500, September 1998.

[106] Dragomir R. Radev, John Prager, and Valerie Samn. Ranking Suspected Answers to Natural Language Questions Using Predictive Annotations. In *The 6th Conference on Applied Natural Language Processing (ANLP)*, Seattle, Washington, 2000.

[107] Ehud Reiter and Somayajulu Sripada. Learning the Meaning and Usage of Time Phrases from a Parallel Text-Data Corpus. In *Proceedings of the HLT-NAACL'03 Workshop on Learning Word Meaning from Non-Linguistic Data*, Edmonton, 2003.

[108] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[109] Gerard Salton, James Allan, and Chris Buckley. Approaches to Passage REtrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, 1993.

[110] Linda Schamber. Relevance and Information Behavior. *Annual Review of Information Science and Technology*, 29:3–48, 1994.

[111] Barry Schiffman. *Learning to Identify New Information.* PhD thesis, 2005.

[112] Frank Schilder and Christopher Habel. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 2001.

[113] John Seely Brown and Paul Duguid. The Social Life of Documents. http://www.firstmonday.dk/issues/issue1/documents/, 1996.

[114] Eugene Seneta. *Non-negative matrices and markov chains.* Springer-Verlag, New York, 1981.

[115] Sidney Siegel and N. John Castellan. *Nonparametric Statistics for the Behavioral Sciences.* McGraw Hill, 1988.

[116] Ian Soboroff and Donna Harman. Overview of the TREC 2003 Novelty Track. In *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, NIST, Gaithersburg, ML, 2003.

[117] Martin M. Soubbotin. Patterns of Potential Answer Expressions as Clues to the Right Answers. In *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 10)*, pages 293–302, Gaithersburg, MD, 2002.

[118] Russell Swan and James Allan. Automatic Generation of Overview Timelines. In *Proceedings of SIGIR '00*, Athens, Greece, 2000.

[119] Russell C. Swan and James Allan. Improving Interactive Information Retrieval Effectiveness with 3-D Graphics. Technical Report IR-100, Department of Computer Science, University of Massachusetts, Amherst, 1996.

[120] Muh-Chyun Tang and Ying Sun. Evaluation of Web-Based Search Engines Using User-Effort Measures. *Library and Information Science Research Electronic Journal*, 13(2), 2003.

[121] Jaime Teevan. *The Re-Search Engine: Helping People Return to Information in Dynamic Information Enviroments*. PhD thesis, 2006.

[122] Simone Teufel and Marc Moens. Articles Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. 28(4), December 2002.

[123] Ndaeyo Uko. Hard News is No News: The Changing Shape of News in the 21st Century. In *Proceedings of the 23rd Annual Conference of the Australia and New Zealand Communication Association (ANZCA 2002)*, Queensland, 2002.

[124] Hans van Halteren and Simone Teufel. Examining the Consensus Between Human Summaries: Initial Experiments with Factoid Analysis. In *Proceedings of HLT-NAACL 2003 Workshop on Text Summarization (DUC03)*, Edmonton, 2003.

[125] Ellen Voorhees and Dawn Tice. The TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, Gaithersburg, MD, 2000.

[126] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Research and Development in Information Retrieval*, pages 315–323, 1998.

[127] William J. Weiland and Ben Schneiderman. A Graphical Query Interface Based on Aggregation/Generalization Hierarchies. *Information Systems*, 18(4):215–232, 1993.

[128] Haris Wu, Dragomir R. Radev, and Weiguo Fan. Towards Answer-Focused Summarization Using Search Engines. In Mark Maybury, editor, *New Directions in Question Answering*, 2003.

[129] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and Redundancy Detection in Adaptive Filtering. In *Proceedings of SIGIR 2002*, 2002.

[130] Zhu Zhang, Sasha Blair-Goldensohn, and Dragomir Radev. Towards CST-enhanced Summarization. In *AAAI 2002 Conference*, Edmonton, Alberta, July-August 2002.

[131] Zhu Zhang, Jahna Otterbacher, and Dragomir Radev. Learning Cross-document Structural Relationships Using Boosting. In *Proceedings of ACM CIKM*, New Orleans, November 2003.

[132] Zhu Zhang and Dragomir Radev. Learning Cross-document Structural Relationships Using Both Labeled and Unlabeled Data. In *Proceedings of IJC-NLP 2004*, Hainan Island, China, March 2004.

[133] Lina Zhou, Judee K. Burgoon, Douglas P. Twitchell, Tiantian Qin, and Jay F. Nunamaker. A Comparison of Classification Methods for Predicting Deception in Computer-Mediated Communication. *Journal of Management Information Systems*, 20(4), 2004.

ABSTRACT

Short-term Event Tracking in Dynamic Online News

by

Jahna Clare Otterbacher

Chair: Dragomir R. Radev

When an important event happens, such as a terrorist attack or natural disaster, many people turn to the World Wide Web to keep track of the most current information. Because large numbers of online agencies report on such events, and continually update their stories, the Web provides timely access to a variety of perspectives. However, following facts in a breaking story is challenging for a number of reasons. For example, news agencies have their own reputation and agenda, such that sources often contradict one another. In addition, it takes time for accounts of stories to stabilize and to be accepted as the ground truth, such that previously reported information is often corrected. Information retrieval applications, such as text summarizers and question answering systems, are designed to help users find relevant information effectively when faced with large amounts of text. However, they typically do not account for the fact that information may be time or source-sensitive. The current thesis works towards designing tools that can support users in following dynamic information, by focusing on the problem of finding facts from sets of related news articles, published while a news story is developing.

Based on the findings of a corpus analysis, as well as an annotation experiment, a prototype system was built. An important finding was that when presented with a factual question and a set of articles about a story, users agreed on which sentences reported answers to the question. However, the agreement as to which answers were new, or had changed with time, was no better than expected by chance. Therefore, rather than detecting changing information, the system finds sentences that are relevant to an input question, and presents them to the user with their respective publication times and sources. The system was evaluated intrinsically and extrinsically with significant results. In particular, in a task-oriented user evaluation, in which the use of the system was compared that of another state-of-the-art system, it was shown that users exerted less effort in searching for the answers to questions with the new system, while obtaining the same level of task accuracy.