TR CUCS-026-99

Topic Shift Detection - Finding New Information in Threaded News

Dragomir R. Radev
Department of Computer Science
Columbia University
New York, NY 10027
radev@cs.columbia.edu

Abstract

On-line sources of news typically follow a particular pattern when presenting updates on a news event over time. First, they produce a preliminary report on the event, and later send out updates as the story evolves. There are two classes of readers accessing the latter stories - these who have read the original announcement and are familiar with the story background and those who are "joining" the thread at a later point in time. Because of the existence of the two clases of readers, news sources typically include in consequent stories some information that was already present in earlier stories. We discuss our approach to identifying such repeated pieces of information in news threads and show how this knowledge can help in generating user-specific summaries of entire threads of articles.

1 Introduction

To be able to generate summaries of threads of articles, it is important to do two things: identify which articles belong together (because they refer to the same event) and identify which portion in subsequent articles contains new information about the event.

We have developed pair of simple algorithms to address these two tasks: a) topic detection (clustering), and b) new information identification. A description of the latest version of our topic detection algorithm can be found in [4]. In the current paper, we present an earlier implementation which we actually use as part of our new information finder (NIF).

Currently, the output of NIF can be used in two ways: as a stand-alone information retrieval system, and more importantly, as a component of SUMMONS. SUMMONS is a knowledge-based text generator which produces summaries of multiple sources [5, 3].

We have also developed a Web-based system which performs the two algorithms above at the user's request. First, it clusters articles from its database into events and then highlights the portions of the articles that present new, old, and background information within the cluster.

2 Motivation

Electronic reports on an event vary along two dimensions: the sources and the time of the report.

Multiple news agencies report on the same event in their news wires. For example, the bombing of the pharmaceutical factory near Khartoum by American missiles in August 1998 was reported electronically by more than two dozen press agencies and newspapers (`http://nt.excite.com/`). A news search gave us 183 articles from sources as diverse as the Bergen Record (New Jersey, USA), the Sydney Morning Herald (Australia), and Radio Free Europe (Prague, Czech Republic).

The articles covered the time range of August 22nd to September 1st, that is 11 days. Earlier articles reported directly about the event, while latter ones mentioned the event in passing. One of the latter articles was a biography of Osama bin Laden who was the alleged reason for the bombing. Another article discussed prospects of international terrorism in the light of the recent events in Kenya and Tanzania and the subsequent bombings in the Sudan and Afghanistan.

The problem of determining which stories in a set of newswire feeds are related to a particular event is called *topic detection and tracking* (TDT) [1]. Our topic detection system, CIDR, is described elsewhere [4]. A more central goal for this paper is to determine which sentences in them present new information. We will use the acronym NIF to refer to the "new information finding" algorithm.

BRAZZAVILLE (Reuter) - A 72-year-old Iranian cyclist touring the world to publicize the plight of children has been stuck in Congo for more than two months after a series of disasters.
KOUROU, French Guiana (Reuter) - Western Europe's 82nd Ariane rocket blasted off into space from French Guiana Friday, putting a U.S. and a Malaysian communications satellite into orbit.

Figure 1: Indication (in bold face) of the location of the report in NANTC.

We should indicate that our algorithm for clustering is quite simple and that it relies on an important assumption: that a time-dependent corpus of news exists in which each story is annotated by the main location where it occurs. We also limited our analysis to locations about which a relatively small number of stories exist.

We will be using the term **time-dependent corpus** to refer to a text corpus in which all documents have a time stamp. Such corpora present interesting properties pertinent to multi-document summarization which we will exploit. More specifically, time-dependent corpora on the same or related events present some degree of redundancy that we exploit in NIF.

3 Our approach to clustering

Traditionally, a large number of different distance measures for clustering of text have been used, such as Euclidean distance, cosine measure, etc. All of them have some advantages and drawbacks. Our task is relatively simple (assuming that the location in which each story takes place and also assuming that only cluster of a specific size will be used), we decided to make use of the fact that a simple heuristic (namely, the use of the main location referred to in an article) gives reasonably good results in clustering news stories so that they can be used by SUMMONS. Later in this paper, we show the results of our experiments.

Since the location is essential to our method, it becomes important to be able to extract it automatically and unambiguously from each article. There are two approaches to this problem - one is to use information extraction, e.g., [2], the other - to use the structure of the actual article. Many news sources include the location of the report at the beginning of the document (see Figure 1). Most of the time, the location of the report can be used as a good approximation of the location in which the event took place. We have actually gone further, using the location of the report as our main heuristic. We have ignored the problem of actually identifying the report location if it is not provided in a trivial manner by the agency. This way, we have decoupled the problem of determining the report location and its use as a heuristic, thus facilitating separate evaluation of the two parts.

We use a modified cosine measure (the inner product of n-dimensional vectors): [8, 7]:

$$SIM(DOC_i, DOC_j) = \frac{\sum_{k=1}^{t}(DOC_{i,k} * DOC_{j,k}) * IDF_k}{\sqrt{\sum_{k=1}^{t}(DOC_{i,k})^2 * \sum_{k=1}^{t}(DOC_{j,k})^2}} \quad (1)$$

where NB_DOCS is the number of documents in the collection and where

$$IDF_k = log(\mathrm{NB\_DOCS}/DF_k) \quad (2)$$

IDF_k is the inverse document frequence of the word k. The equations are based on the cosine formula:

$$cos\gamma = \frac{A.B}{||A||.||B||} \quad (3)$$

We consider two articles to be on the same event if their similarity (SIM) is above a certain pre-defined threshold.

4 Experiments and results

For our experiments, we picked a subset of the NANTC corpus[1]. It contains news from Reuters, the New York Times, and several other sources. We performed most of the experiments on the 8,607 articles and 628 locations from January 1996 that originated from Reuters. Table 1 shows the distribution of stories by location.

The most frequently encountered cities are shown in Table 2. However, these cities contained hardly any articles on terrorism, so we didn't use them in our evaluation.

In our first experiment, we manually split the 24 stories located in Berlin into clusters. Our clustering is shown in Table 3 and Figure 2.

[1] It was made available to us by the Linguistic Data Consortium.

Location	**Number of stories**	**Distribution by day of the month**
Aberdeen	1	0000000000000000000000000000100
Abidjan	10	0010200000000000000000220002010
Abuja	2	0000000000000000000100000100000
Accra	1	0000000001000000000000000000000
Addis Ababa	7	0000000000000002002000000000021
Adelaide	3	0001001000000010000000000000000
Ajaccio	22	4001102232231000000010000000000
Aksai	1	0000000000000000010000000000000
Albany	1	0001000000000000000000000000000
Almaty	16	0002000000000000000004101304010
Alvord	5	0000000000050000000000000000000
Amherst	1	1000000000000000000000000000000
Amman	28	0300005353100200100002100020000
Amsterdam	11	1101000111000001100000110100000
...		...
Berlin	24	1000000022111050120000001000034
...		...
Karachi	20	5200003031000000000400002000000
...		...
Kigali	14	0000000000000001300312011000002
...		...
Lima	29	0101000310500012230101400100021
...		...
Sofia	10	0000110102000000001001210000000
...		...
Rio De Janeiro	23	0001012311000011112212201000000

Table 1: Distribution of stories by location in January of 1996.

Location	Number of articles
Washington	1003
Moscow	320
London	306
New York	280
Paris	275
Jerusalem	234
Beijing	228
Tokyo	221

Table 2: Most frequently encountered locations in January of 1996.

Story No.	Story ID	topic
1	BERLIN/960101.0073	firework deaths
2	BERLIN/960109.0101	Vogel trial
3	BERLIN/960109.0201	Vogel trial
4	BERLIN/960110.0288	Vogel trial
5	BERLIN/960110.0292	Free Democrats
6	BERLIN/960111.0320	Iranian secret service
7	BERLIN/960112.0193	Berlin coalition
8	BERLIN/960113.0070	Free Democrats
9	BERLIN/960115.0059	Krenz trial
10	BERLIN/960115.0092	Weizman visit
11	BERLIN/960115.0128	Krenz trial
12	BERLIN/960115.0165	Krenz trial
13	BERLIN/960115.0193	Weizman visit
14	BERLIN/960117.0297	Vogel trial
15	BERLIN/960118.0079	Berlin coalition
16	BERLIN/960118.0229	Berlin coalition
17	BERLIN/960125.0235	Iranian secret service
18	BERLIN/960130.0126	Schnur trial
19	BERLIN/960130.0200	Greenpeace
20	BERLIN/960130.0206	Schnur trial
21	BERLIN/960131.0087	Schalck-Golodkowski trial
22	BERLIN/960131.0135	Schalck-Golodkowski trial
23	BERLIN/960131.0165	Schalck-Golodkowski trial
24	BERLIN/960131.0242	Schalck-Golodkowski trial

Table 3: Correct distribution of stories located in Berlin.

The values of SIM for BERLIN are shown in Tables 4 and 5. Values above the threshold are marked by a rectangle around the similarity value.

Table 6 shows the evaluation of system performance for the Berlin stories. The actual stories that form cluster number 2 (stories with numbers 2, 3, 4, and 14) are shown in [3].

NIF achieves 95.83% precision and 95.83% recall on the BERLIN cluster (Figure 3). The average precision and recall over all cases in our small-scale experiment are 84.62% precision and 84.62% recall.

Table 7 shows the precision and recall values for four randomly chosen cities among those with fewer than 100 articles: Berlin, Sofia, Lima, and Reykjavik.

5 Web interface

NIF1 has a stand-alone Web interface, a snapshot of which is shown in Figure 4. The user can specify which location he is interested in and see how the clusters of news stories are distributed by topics over the selected period of time.

6 Identifying new and old information in clusters of news

We mentioned earlier that that often news writers repeat a large amount of information from one story to another. For example, Figures 5 and 6 show excerpts from two articles that were found to be in the same cluster by the module described in the previous sections. The figures show the two paragraphs of the first story and the first five paragraphs of the second story (out of 18).

One can notice that paragraphs 1 and 3 in the second story essentially convey the same information as paragraphs 1 and 2 in the first story, respectively. There are at least three reasons why this happens in news writing:

- when the earlier story served the purpose of breaking urgent news and the details are written in a follow-up story.
- when the second story serves as a background to the first one.
- when the latter story adds new information to the story while keeping the user informed about earlier developments.

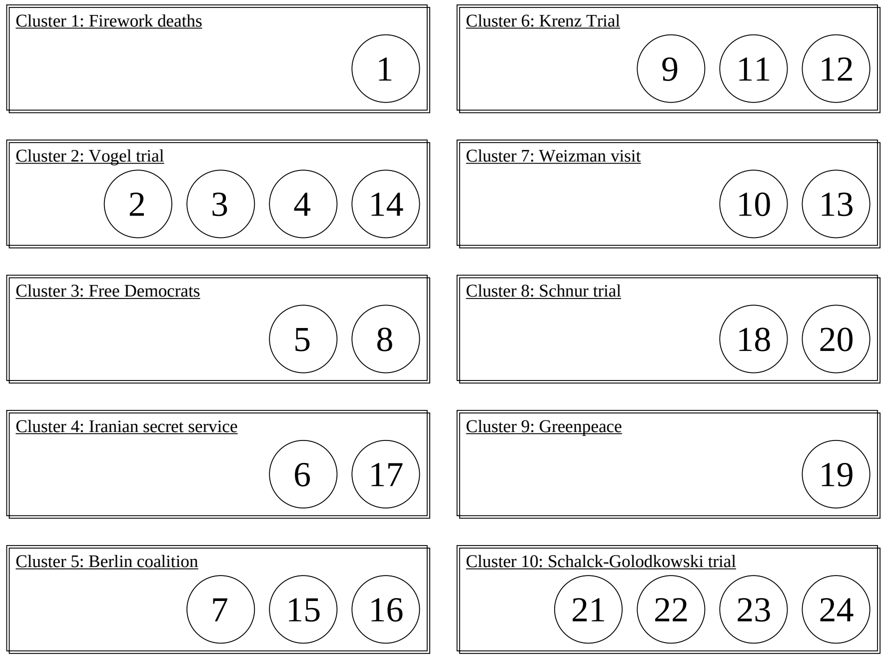

Figure 2: Correct assignment for the BERLIN cluster.

	1	2	3	4	5	6	7	8	9	10	11	12
1	**1.00**	0.18	0.08	0.11	0.06	0.11	0.22	0.09	0.15	0.15	0.15	0.10
2	0.18	**1.00**	**0.59**	**0.57**	0.07	0.17	0.32	0.13	0.37	0.17	0.37	0.18
3	0.08	**0.59**	**1.00**	**0.69**	0.06	0.08	0.14	0.05	0.33	0.08	0.33	0.17
4	0.11	**0.57**	**0.69**	**1.00**	0.06	0.12	0.20	0.07	0.37	0.11	0.37	0.20
5	0.06	0.07	0.06	0.06	**1.00**	0.05	0.13	**0.63**	0.10	0.08	0.10	0.09
6	0.11	0.17	0.08	0.12	0.05	**1.00**	0.19	0.08	0.15	0.10	0.15	0.12
7	0.22	0.32	0.14	0.20	0.13	0.19	**1.00**	0.20	0.26	0.22	0.26	0.20
8	0.09	0.13	0.05	0.07	**0.63**	0.08	0.20	**1.00**	0.12	0.10	0.12	0.09
9	0.15	0.37	0.33	0.37	0.10	0.15	0.26	0.12	**1.00**	0.15	**0.99**	**0.79**
10	0.15	0.17	0.08	0.11	0.08	0.10	0.22	0.10	0.15	**1.00**	0.15	0.10
11	0.15	0.37	0.33	0.37	0.10	0.15	0.26	0.12	**0.99**	0.15	**1.00**	**0.80**
12	0.10	0.18	0.17	0.20	0.09	0.12	0.20	0.09	**0.79**	0.10	**0.80**	**1.00**
13	0.12	0.13	0.08	0.10	0.09	0.09	0.17	0.08	0.14	**0.86**	0.14	0.11
14	0.13	**0.57**	**0.65**	**0.87**	0.07	0.13	0.23	0.09	0.36	0.12	0.36	0.19
15	0.16	0.21	0.08	0.14	0.11	0.14	**0.63**	0.16	0.19	0.17	0.19	0.15
16	0.16	0.21	0.08	0.14	0.11	0.14	**0.63**	0.16	0.19	0.17	0.19	0.15
17	0.15	0.22	0.12	0.15	0.10	0.55	0.25	0.11	0.21	0.15	0.21	0.17
18	0.06	0.13	0.09	0.13	0.12	0.07	0.15	0.10	0.15	0.08	0.15	0.11
19	0.06	0.09	0.06	0.05	0.06	0.05	0.06	0.04	0.08	0.10	0.08	0.05
20	0.07	0.13	0.10	0.14	0.12	0.07	0.15	0.10	0.15	0.09	0.15	0.12
21	0.10	0.20	0.14	0.18	0.09	0.09	0.18	0.07	0.18	0.10	0.18	0.15
22	0.08	0.26	0.29	0.29	0.08	0.08	0.11	0.04	0.27	0.08	0.28	0.21
23	0.08	0.28	0.27	0.28	0.09	0.09	0.14	0.05	0.22	0.09	0.23	0.19
24	0.09	0.28	0.30	0.29	0.08	0.09	0.14	0.06	0.28	0.09	0.28	0.21

Table 4: Similarities among the BERLIN articles (Part 1).

	13	14	15	16	17	18	19	20	21	22	23	24
1	0.12	0.13	0.16	0.16	0.15	0.06	0.06	0.07	0.10	0.08	0.08	0.09
2	0.13	0.57	0.21	0.21	0.22	0.13	0.09	0.13	0.20	0.26	0.28	0.28
3	0.08	0.65	0.08	0.08	0.12	0.09	0.06	0.10	0.14	0.29	0.27	0.30
4	0.10	0.87	0.14	0.14	0.15	0.13	0.05	0.14	0.18	0.29	0.28	0.29
5	0.09	0.07	0.11	0.11	0.10	0.12	0.06	0.12	0.09	0.08	0.09	0.08
6	0.09	0.13	0.14	0.14	0.55	0.07	0.05	0.07	0.09	0.08	0.09	0.09
7	0.17	0.23	0.63	0.63	0.25	0.15	0.06	0.15	0.18	0.11	0.14	0.14
8	0.08	0.09	0.16	0.16	0.11	0.10	0.04	0.10	0.07	0.04	0.05	0.06
9	0.14	0.36	0.19	0.19	0.21	0.15	0.08	0.15	0.18	0.27	0.22	0.28
10	0.86	0.12	0.17	0.17	0.15	0.08	0.10	0.09	0.10	0.08	0.09	0.09
11	0.14	0.36	0.19	0.19	0.21	0.15	0.08	0.15	0.18	0.28	0.23	0.28
12	0.11	0.19	0.15	0.15	0.17	0.11	0.05	0.12	0.15	0.21	0.19	0.21
13	1.00	0.11	0.15	0.15	0.15	0.09	0.10	0.10	0.13	0.10	0.13	0.11
14	0.11	1.00	0.17	0.17	0.17	0.12	0.06	0.12	0.21	0.28	0.28	0.29
15	0.15	0.17	1.00	1.00	0.19	0.17	0.05	0.17	0.14	0.08	0.11	0.10
16	0.15	0.17	1.00	1.00	0.19	0.17	0.05	0.17	0.14	0.08	0.11	0.10
17	0.15	0.17	0.19	0.19	1.00	0.11	0.08	0.11	0.16	0.14	0.17	0.14
18	0.09	0.12	0.17	0.17	0.11	1.00	0.04	1.00	0.11	0.14	0.17	0.12
19	0.10	0.06	0.05	0.05	0.08	0.04	1.00	0.04	0.05	0.06	0.06	0.06
20	0.10	0.12	0.17	0.17	0.11	1.00	0.04	1.00	0.12	0.15	0.17	0.12
21	0.13	0.21	0.14	0.14	0.16	0.11	0.05	0.12	1.00	0.63	0.65	0.59
22	0.10	0.28	0.08	0.08	0.14	0.14	0.06	0.15	0.63	1.00	0.88	0.86
23	0.13	0.28	0.11	0.11	0.17	0.17	0.06	0.17	0.65	0.88	1.00	0.73
24	0.11	0.29	0.10	0.10	0.14	0.12	0.06	0.12	0.59	0.86	0.73	1.00

Table 5: Similarities among the BERLIN articles (Part 2).

Model	Partition
1. 1	1. 1
2. 2 3 4 14	2. 2 3 4 14
3. 5 8	3. 5 8
4. [6]	4. [6 17]
5. 7 15 16	5. 7 15 16
6. 9 11 12	6. 9 11 12
7. 10 13	7. 10 13
8. [17]	8. []
9. 18 20	9. 18 20
10. 19	10. 19
11. 21 22 23 24	11. 21 22 23 24

Figure 3: Evaluation of clustering. Partition (system output) is compared against the Model.

Cluster No.	Cluster size	Precision	Cluster size	Recall
1	1	100.00 %	1	100.00 %
2	4	100.00 %	4	100.00 %
3	2	100.00 %	2	100.00 %
4	2	50.00 %	1	100.00 %
5	3	100.00 %	3	100.00 %
6	3	100.00 %	3	100.00 %
7	2	100.00 %	2	100.00 %
8	0	100.00 %	1	0.00 %
9	2	100.00 %	2	100.00 %
10	1	100.00 %	1	100.00 %
11	4	100.00 %	4	100.00 %
average		95.83 %		95.83 %

Table 6: Performance of NIF on a single location (BERLIN).

City	Number of stories	Precision	Recall
Berlin	24	95.83%	95.83%
Sofia	10	90.00%	90.00%
Lima	29	72.41%	72.41%
Reykjavik	2	100.00%	100.00%
average	65	84.62%	84.62%

Table 7: Four-city performance.

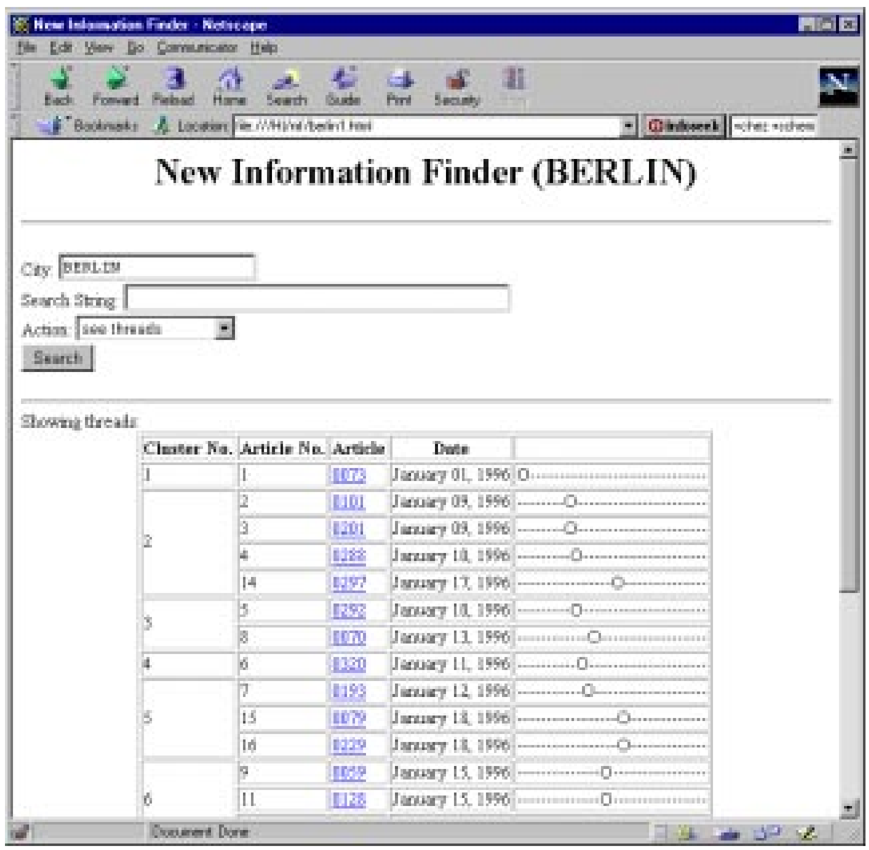

Figure 4: Web-based interface to NIF.

```
<DOCID> reute960109.0101 </DOCID>

...

<HEADER> reute 01-09 0057 </HEADER>

...

 German court convicts Vogel of extortion

    BERLIN, Jan 9 (Reuter) - A German court on Tuesday convicted
Wolfgang Vogel, the East Berlin lawyer famous for organising
Cold War spy swaps, on charges that he extorted money from
would-be East German emigrants.

    The Berlin court gave him a two-year suspended jail sentence
and a fine -- less than the 3 3/8 years prosecutors had sought.
```

Figure 5: Two paragraphs from the first story in the BERLIN cluster.

When news journalists know that *all* potential readers would have enough background on the event they do not repeat the background information. For example, because of the popularity of the Clinton/Lewinsky scandal, latter stories rarely described how the entire thing started. However, stories about developments on less talked about topics such as the Swissair Flight 111 crash and the bombings in Kenya and Tanzania typically included some information about the background of the story.

In generating summaries of clusters of articles on the same topic, one would obviously run across cases of repeated information. Again, if the summarizer keeps track of its interaction with a particular user, it doesn't need to include any information in the later summaries if that information has already been used in earlier summaries. We call this setup an **evolving summary** and we will spend the rest of this paper discussing some techniques that can be used to produce evolving summaries.

Definition 1 *An* **evolving summary** *is the summary of a story, numbered A_{k+1} when the stories numbered A_1 to A_k have already been processed and presented in a summarized form to the user.*

At this point, we would like to note that being able to identify new and repeated information in clusters of stories can be helpful for both statistical and conceptual summarizers:

Statistical summarizers

Sentences that contain repeated information should be ignored or assigned low scores prior to sentence extraction. Our analysis shows that most of the repeated sentences appear in the first 2-3 paragraphs of a new story. Given that [6] had suggested that these are the paragraphs that should be assigned the highest scores, it is obvious that the ability to weed out such sentences will help produce better evolving summaries.

Similarly, being able to identify new vs. background information can help in producing better briefings (remember that briefings are defined to ignore background information).

Conceptual summarizers

The advantages of recognizing repeated information is not limited to sentence extraction. In the SUMMONS paradigm, one could run the MUC system only on text that has not been labeled as repeated.

7 Finding related paragraphs in threads

We have identified four classes of sentences (paragraphs) according to their purpose:

- **N**: New (breaking/current) information : e.g., the announcement of a plane crash right after the accident.

```
<DOCID> reute960109.0201 </DOCID>

...

<HEADER> reute 01-09 0582 </HEADER>

...

 East German spy-swap lawyer convicted of extortion

    BERLIN (Reuter) - The East Berlin lawyer who became famous
for engineering Cold War spy swaps, Wolfgang Vogel, was
convicted by a German court Tuesday of extorting money from East
German emigrants eager to flee to the West.

    Vogel, a close confidant of former East German leader Erich
Honecker and one of the Soviet bloc's rare millionaires, was
found guilty of perjury, four counts of blackmail and five
counts of falsifying documents.

    The Berlin court gave him the two-year suspended sentence
and a $63,500 fine. Prosecutors had pressed for a jail sentence
of 3 3/8 years and a $215,000 penalty.

    Vogel, 70, who got his start arranging the 1962 exchange of
U.S. pilot Gary Powers for Soviet spy Rudolf Abel, insisted his
only crime was trying to help unite people separated by the Cold
War division of Germany.

    ``The court said that I helped people -- what more can I
say?'' Vogel said after Judge Heinz Holzinger spent 90 minutes
reading the verdict to a packed courtroom.
```

Figure 6: The first five paragraphs from the second story in the BERLIN cluster.

Original	Copies
1	3 21 28
2	5 26 32
4	25 31
6	27 35
10	23

Table 8: System output on the Berlin cluster.

- **B**: Background information: e.g., a history of prior crashes by planes of the same company.
- **R**: Repeated information: e.g., a mention of the fact that the plane crashed appearing in subsequent stories which are primarily concerned with describing the development of the salvage operation.
- **O**: Other: in this class, we group anecdotal leads and quotes from participants in the investigation, as well as any other sentence not categorized in either the **N**, **B**, and **R** classes.

We will refer to these four classes as the **purpose** of the sentences that they categorize.
For the purpose of creating evolving summaries we decided that four problems are worth investigating:

- **N-type recognition**: highest priority - these sentences (or information extracted from them, in the case of conceptual summarization) should appear in the summary with the highest priority.
- **B-type recognition**: sentences of this class will be assigned low priority before summarizing the story that contains them.
- **R-type recognition**: these should not be processed if the system knows that the user has already seen summaries produced based on the earlier instances of related sentences.
- **O-type recognition**: we consider these sentences the least important to summarization.

We decided to focus on the fourth of these problems - the binary classification of paragraphs in clusters into R-type and not-R-type paragraphs. For this purpose, we annotated manually a corpus of clusters of news stories and used a portion of it for developing a method for R-type labeling. We used the rest of the corpus (unseen during training) for evaluation.

8 Methodology

Our initial thought was to focus on primarily linguistic and stylistic features (such as the presence of quotes and proper nouns in different paragraphs). However, after a few experiments, we discovered that a simple statistical method, similar to the one that we used in the previous sections for the clustering itself, achieves the best results.

We already described the algorithm that we use to cluster articles together. We use the same algorithm (at the paragraph level) to identify related paragraphs in entire threads of article.

For illustration of our approach, we will use the four stories in the cluster about Berlin (we remind the reader that NIF1 was used for the actual clustering). The number of paragraphs in the four stories are 2, 18, 7, and 8, respectively.

For the rest of this paper we will refer to each group of related paragraphs within a cluster as a **group of related paragraphs**. The first paragraph (chronologically) in a group will be called the **original** while the remaining ones will be referred to as the **copies** of the original. Of course, these paragraphs are not identical copies of the original, they are simply highly similar to it.

When we ran our algorithm on the Berlin cluster, we obtained 24 groups of related paragraphs. Obviously, the first paragraph of each group (also 24 in total) is labeled as not-R-type, while the remaining 11 paragraphs are marked to be of R-type. The partition and model comparison is displayed in Table 8. Table 9 shows the contingency table used to measure precision and recall for R-type classification in the Berlin example. The corresponding precision is $10/11 = 90.9\%$ and recall – $9/10 = 90.00\%$.

9 Conclusion

This paper discusses a property of news threads - the fact that latter stories in a thread on a given event often contain repeated information which is unnecessary for the reader if he has already read the previous stories in the

		Partition	
		R-type	**not-R-type**
Model	**R-type**	9	2
	non-R-type	1	23

Table 9: Evaluation of R-type recall and precision in the Berlin cluster.

thread. We discuss a) our approach to the automatic creation of threads of news on the same event based on the location of the report, and b) a technique for identifying repeated paragraphs in news threads. We also discuss how the knowledge of such repeated information can be used to improve the operation of both knowledge-based and sentence-extraction based summarizers.

10 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. IRI-96-19124, IRI-96-18797, and CDA-96-25374, as well as a grant from Columbia University's Strategic Initiative Fund sponsored by the Provost's Office. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

The author is grateful to the following people for their comments and suggestions: Kathy McKeown, Vasileios Hatzivassiloglou, Al Aho, and Hongyan Jing from Columbia, Ed Hovy from USC/ISI, and Julian Kupiec from Xerox PARC.

References

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

[2] Wendy Lehnert, Joe McCarthy, Stephen Soderland, Ellen Riloff, Claire Cardie, Jonathan Peterson, and Fangfang Feng. UMass/Hughes: Description of the CIRCUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 277–291, Baltimore, Md., August 1993.

[3] Dragomir R. Radev. *Language Reuse and Regeneration: Generating Natural Language Summaries from Multiple On-Line Sources.* PhD thesis, Department of Computer Science, Columbia University, New York, April 1999.

[4] Dragomir R. Radev, Vasileios Hatzivassiloglou, and Kathleen R. McKeown. A description of the CIDR system as used for TDT-2. In *DARPA Broadcast News Workshop*, Herndon, VA, February 1999.

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

[6] Lisa F. Rau, Ron Brandow, and Karl Mitze. Domain-Independent summarization of news. In *Summarizing Text for Intelligent Communication*, pages 71–75, Dagstuhl, Germany, 1994.

[7] Gerard Salton. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, 1989.

[8] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* Computer Series. McGraw Hill, New York, 1983.