INCIDENT THREADING IN NEWS

A Dissertation Presented

by

AO FENG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2008

Computer Science

© Copyright by Ao Feng 2008
All Rights Reserved

INCIDENT THREADING IN NEWS

A Dissertation Presented

by

Ao Feng

Approved as to style and content by:

____
James Allan, Chair

____
W. Bruce Croft, Member

____
Paul E. Utgoff, Member

____
John Staudenmayer, Member

____
R. Manmatha, Member

____
Andrew G. Barto, Department Head
Computer Science

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-002. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

ABSTRACT

INCIDENT THREADING IN NEWS

September 2008

AO FENG, B.Eng., TSINGHUA UNIVERSITY

M.Eng., TSINGHUA UNIVERSITY

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

With an overwhelming volume of news reports currently available, there is an increasing need for automatic techniques to analyze and present news to a general reader in a meaningful and efficient manner. Previous research has focused primarily on organizing news stories into a list of clusters by the main topics that they discuss. We believe that viewing a news topic as a simple collection of stories is restrictive and inefficient for a user hoping to understand the information quickly.

As a proposed solution to the automatic news organization problem, we introduce *incident threading* in this thesis. All text that describes the occurrence of a real-world happening is merged into a news incident, and incidents are organized in a network with dependencies of predefined types.

In order to simplify the implementation, we start with the common assumption that a news story is coherent in content. In the *story threading* system, a cluster of news documents discussing the same topic are further grouped into smaller sets, where each represents a separate news event. Binary links are established to reflect the contextual information among those events. Experiments in story threading show promising results. We next describe an enhanced version called *relation-oriented story threading* that extends the range of the prior work by assigning type labels to the links and describing the relation within each story pair as a competitive process among multiple options. The quality of links is greatly improved with a global optimization process.

Our final approach, *passage threading*, removes the story-coherence assumption by conducting passage-level processing of news. First we develop a new testbed for this research and extend the evaluation methods to address new issues. Next, a calibration study demonstrates that an incident network helps reading comprehension with an accuracy of 25-30% in a matrix comparison evaluation. Then a new three-stage algorithm is described that identifies on-subject passages, groups them into incidents, and establishes links between related incidents. Finally, significant improvement over earlier work is observed when the training phase optimizes the harmonic mean of various evaluation measures, and the performance meets the goal in the calibration study.

TABLE OF CONTENTS

Page

CHAPTER

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the fast development of modern technologies, the amount of accessible information is increasing in an exponential manner [O'Leary 1997]. Every day there is a large amount of new information available to us, and much of it is news. It comes from many different sources, including traditional media such as newspaper, radio and TV, and modern sources like the World Wide Web (WWW). Without proper arrangement of the overwhelming information, one can easily become lost because of its vast size. This phenomenon is called *information overload*.

It is not feasible for a user to go through all the information without any pre-processing, because the news a person can read in a certain time is much less than the amount that is generated within the same period. To help the user obtain the necessary information in the shortest time, a system is desired that automatically processes news and converts it to a more user-efficient format.

People have their own ways of comprehending news information, but there are some common rules that most would follow. In order for an automatic system to facilitate users effectively in their reading process, it is recommended that this system have similar abilities:

- Each user has his or her own information need. For example, a resident of New York City might be interested in a crime that happened in the City, but may not care if there is a conflict in Kosovo. Therefore, there is little value if news reports from both topics are mixed together. An ideal system should group news according to the main topic discussed.

- People remember interesting information for a long time, and are interested in new messages rather than repetitions, even if the repeated information is described in a different way. It is not advisable that the system provide duplicate information.
- Since human beings have reasoning abilities, they do not treat news events as isolated facts. Instead, they tend to compare new information to memory and insert it into the existing fact network, at a location next to the relevant pieces. It would be ideal if the system has the same ability to link related events, because people are very likely to be interested in both (or neither). In addition, tracing back from the new information can be a good reminder to readers for things that they have already forgotten.

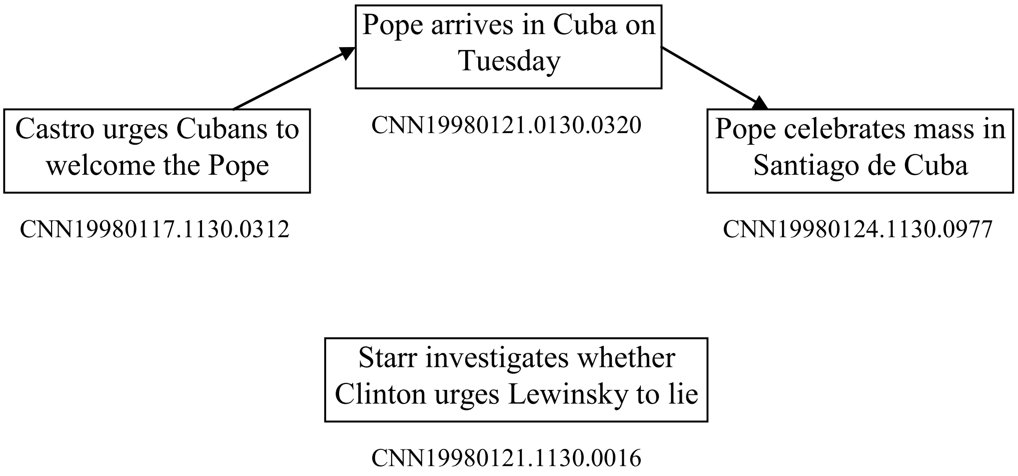

Pope arrives in Cuba on Tuesday
CNN19980121.0130.0320
Castro urges Cubans to welcome the Pope
CNN19980117.1130.0312
Pope celebrates mass in Santiago de Cuba
CNN19980124.1130.0977
Starr investigates whether Clinton urges Lewinsky to lie
CNN19980121.1130.0016

Figure 1: Sample News Reports from CNN

Figure 1 shows summaries of four news reports from CNN. The strings below the boxes are document identifiers from the corpora used in this work. As we can see, three of them are from the same news topic “Pope visits Cuba,” and the last one is about the well-known Monica Lewinsky case. An ideal news organization should place the three related reports together and show their contextual link, leaving the irrelevant information aside. Figure 1 shows such an organization.

In this thesis we introduce a technique called *incident threading*. One of its main goals is to identify all text that discusses the same real-world event and assign the difference pieces to a news incident. Incidents are not treated as isolated entities, because there are links between real-world events, and people have the ability to recognize them. In incident threading, we strive to find causal, temporal, spatial, or other types of connections among the incidents. With these links as edges, the news events form an interconnected network that shows how messages are related. This system is called "incident threading" because the edges in the incident network form information *threads* that show the evolution of news reports.

The most important concepts in the model are *incidents* and *incident networks*. An incident is a news event that happens at a specific time point (or within a certain period of time), at a given geographical location, and involves one or more entities and some action. An incident can appear in various news reports, and the descriptions may not look similar, but it is always the same incident if they talk about the same thing.

News is not static, as there are often new updates on a certain topic. The updates do not belong to the old incident, since the time and/or other factors have changed, but there is some intrinsic connection that links them together. An incident network is a graph that shows these related incidents along with the connections.

Next we illustrate the basic concepts and show how news is organized in this framework with some examples.

1.1 Examples

> (a) JERUSALEM -- The Lebanese guerrilla group Hezbollah surprised Israel with a daylight assault across the border on Wednesday, leading to fighting in which two Israeli soldiers were captured and at least eight killed, and elevating recent tensions into a serious two-front battle.
> (b) Israel, already waging a military operation in the Gaza Strip to free a soldier captured by Palestinian militants on June 25, immediately responded by sending armored forces into southern Lebanon for the first time in six years and holding Lebanon's government responsible for the Hezbollah assault.
> (c) The toll was the highest one for the Israeli soldiers in several years, and combined with the deaths on Wednesday of more than 20 Palestinians, including many civilians, in fighting in Gaza, it was the deadliest day in the Arab-Israeli conflict since Israel withdrew from the Gaza Strip last year. Andthe violence continued into the early morning hours, when an Israeli airstrike heavily damaged the Palestinian Foreign Ministry building in Gaza.
> (d) Even though Israel has military superiority in southern Lebanon and Gaza, the new fighting signaled the emergence of a conflict that has blown past the limits of local confrontation into a regional crisis.
> (e) And some analysts suggested that the similarity between the Hezbollah raid and the earlier one in Gaza by fighters with the Islamic faction Hamas and its allies, both intended to gain leverage through captured Israeli soldiers, may demonstrate a growing and troubling rapport between the two groups.
> (f) As with the Gaza conflict, Israel ruled out negotiations with the Lebanese captors of the Israeli soldiers. Prime Minister Ehud Olmert said he held the Lebanese government responsible for the assault by Hezbollah, the Shiite Muslim group that participates in Lebanese politics but also continues to battle Israel.
> (g) "I want to make clear that the event this morning is not a terror act, but an act of a sovereign state that attacked Israel without reason," Olmert said. "The government of Lebanon, of which Hezbollah is a part, is trying to shake the stability of the region."
> (h) Israel is demanding that its soldiers be returned unconditionally and that militant groups stop firing rockets at Israeli civilians from Gaza in the south and Lebanon in the north.
> (i) But both Hamas and Hezbollah are holding out for an exchange for a large number of Palestinian and other Arab prisoners held by Israel.
> (j) "The prisoners will not be returned except through one way -- indirect negotiations and a trade," said the leader of Hezbollah, Sheik Hassan Nasrallah, speaking to reporters in Beirut late on Wednesday.

Figure 2: Sample News Story - NYT_ENG_20060712.0202 (paragraph segmentation and lettering inserted)

Figure 2 shows a news story with paragraphs marked by letters. The story is about a surprise attack of Hezbollah, a Lebanese guerrilla group, towards Israel. In response, Israel sends troops to southern Lebanon. Paragraph (a) describes the surprise attack. The next paragraph (b) says that Israel is sending troops into Lebanon for retaliation. (c) provides an evaluation for the surprise attack and mentions an Israeli air strike in Palestine. (d) and (e) contain more information about the conflicts. Paragraph (f) describes the reaction of the Israeli government with Prime Minister Ehud Olmert's speech, while (g) continues on it. Paragraph (h) talks about the request of Israel, with the same tone as in Olmert's speech. Paragraph (i) moves the focus to the other side, introducing the aim of Hamas and Hezbollah, and (j) quotes the leader of Hezbollah that the prisoners are for exchange. Here are the incidents in this story and their main features:

1. (a): Hezbollah conducts surprise attack towards Israel. (when: Wednesday; where: border; who: Hezbollah; what: assault)
2. (b): Israel sends troops to southern Lebanon. (when: immediately; where: southern Lebanon; who: Israel; what: send)
3. (c): Israel bombs the Palestinian Foreign Ministry building. (when: early morning; where: Gaza; who: Israel; what: air strike)
4. (f), (g) and (h): Israel refuses negotiations with Hezbollah. (when: N/A; where: N/A; who: Israel; what: rule out)
5. (i) and (j): Hamas and Hezbollah request for a prisoner exchange. (when: N/A; where: N/A; who: Hamas and Hezbollah; what: hold out)

Note that paragraphs (d) and (e) provide some facts and analysis related to this conflict, but they do not directly describe any event that actually happened, so they are not considered as part of any incident.

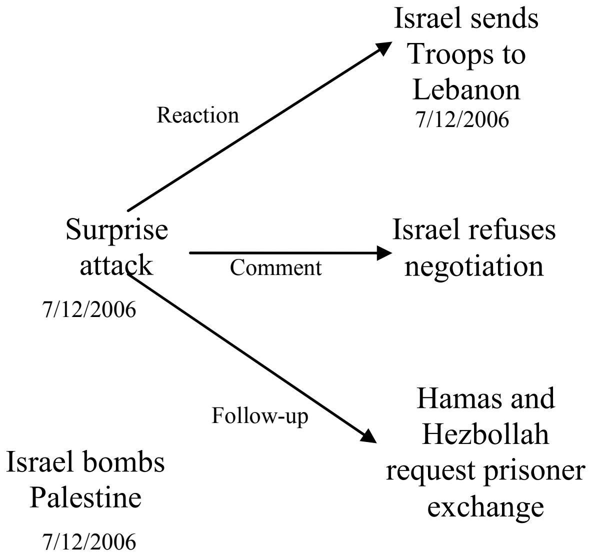

Israel sends Troops to Lebanon
7/12/2006
Reaction
Surprise attack
7/12/2006
Comment
Israel refuses negotiation
Follow-up
Hamas and Hezbollah request prisoner exchange
Israel bombs Palestine
7/12/2006

Figure 3: Incident Network for the Story in Figure 2

Figure 3 displays an incident network formed by the information in that story. A node in the network is an incident, while the text under each node indicates the time when the described event happened. An edge in the figure shows that there exists some relation between two incidents, with the text next to it describing the corresponding link type, which has a limited vocabulary of types.

Information about this conflict does not end here. On the next day, another news story reports the escalation of the conflict, including air bombing conducted by the Israeli army and rockets fired by Hezbollah. This new report is shown in Figure 4. Some facts have already been mentioned in the previous story, like the surprise attack by Hezbollah. But it also contains new information – bombing from Israel and rockets fired by Hezbollah. There are some other incidents in the second story, mainly the damages caused by the escalated conflict. In addition to the direct description of violent activities, the later part of the story (omitted) introduces the statements made by the parties involved in the conflict and comments from other nations.

> (a) ROSH PINA, Israel - Israeli forces struck at Lebanon's air, sea, and land routes yesterday, and fighters from the Lebanese militant group Hezbollah fired more than 100 rockets into Israel in an escalating conflict that left dozens dead and threatened to engulf countries across the Middle East.
> (b) Hezbollah rockets fired from Lebanon reached deeper into Israel than ever before. Two of them hit the Mediterranean port of Haifa, Israel's third-largest city, about 30 miles from the border.
> (c) Israel bombed Beirut's airport, shutting it down, imposed an air and sea blockade on the country, and struck Lebanese Army bases. Last night, Israeli warplanes hit the highway linking Beirut to the Syrian capital of Damascus, further isolating Lebanon from the outside world.
> (d) Early today, Israeli jets struck Hezbollah's stronghold neighborhoods in south Beirut. Israel had dropped leaflets in the area yesterday, urging residents to leave.
> (e) At least 53 Lebanese civilians were killed in strikes yesterday, according to the Lebanese government, and Hezbollah missiles killed two Israeli civilians, the military said. More than 100 were wounded on each side.
> (f) The fighting raised fears of a larger regional conflict, as Israeli officials continued to blame Iran and Syria for their longtime financial and political support of Hezbollah. Yesterday, they declared that Hezbollah might try to spirit two Israeli soldiers it captured in a cross-border raid Wednesday into Iran, although they offered no evidence.
> (g) Israeli officials have said that the country is fighting against a coalition of Islamist groups opposed to the existence of the Jewish state, along with the governments that back them.
> (h) On Wednesday Israel said it held the Lebanese government, in which Hezbollah has two Cabinet seats, responsible for the attack. Israel blames the Palestinian Authority for a similar raid on June 25 in which Palestinian gunmen, including some from the governing Hamas movement, infiltrated Israel from the Gaza Strip and abducted an Israeli soldier.
> (i) (j) (k) …

Figure 4: Another Story in the Same News Topic - NYT_ENG_20060713.0300 (some text omitted)

With the new information from the second story, the network of the current incidents is shown in Figure 5. The node in the dashed box on the left side of the figure is duplicate information that appears in both stories. The user already knows that after reading the first story, and would like to see only the new messages (text in the solid rectangles) when another news report comes in. An incident outside both boxes appears only in the first story, and acts as a reminder of the old information if the reader does not remember it. In Figure 5, the user should be happy to see the follow-up if he/she shows interest at the beginning of the conflict, and the network structure clearly demonstrates the evolution of the topic.

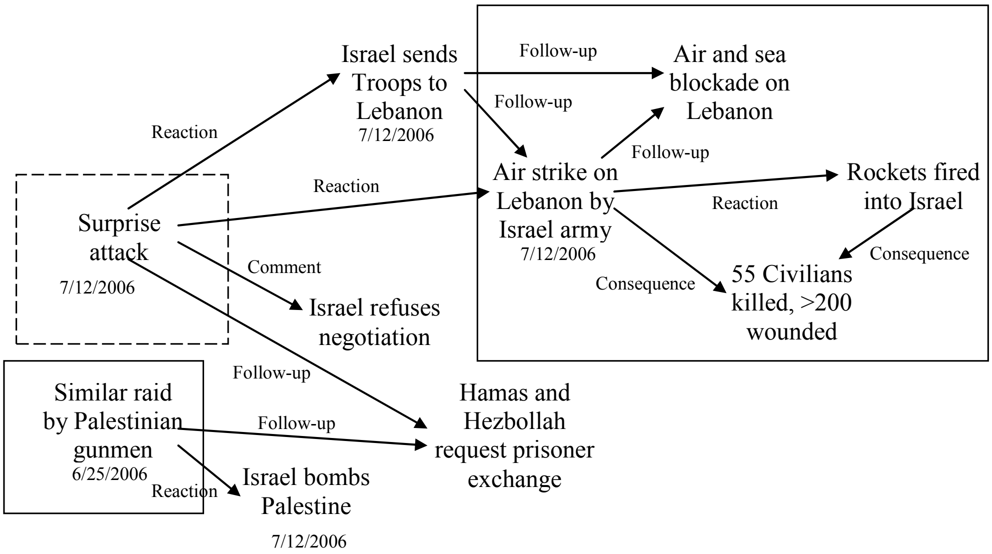

Israel sends Troops to Lebanon
7/12/2006
Follow-up
Air and sea blockade on Lebanon
Follow-up
Reaction
Follow-up
Air strike on Lebanon by Israel army
7/12/2006
Reaction
Rockets fired into Israel
Reaction
Surprise attack
7/12/2006
Comment
Consequence
Consequence
55 Civilians killed, >200 wounded
Israel refuses negotiation
Follow-up
Similar raid by Palestinian gunmen
6/25/2006
Follow-up
Hamas and Hezbollah request prisoner exchange
Reaction
Israel bombs Palestine
7/12/2006

Figure 5: Incident Network for the Stories in Figures 2 and 4

New York Times does not provide more reports about this conflict for a few days. Then on July 17th, 2006, another news story (NYT_ENG_20060717.0240) updates the death toll to 233 with the conflicts in the next days and introduces the diplomatic efforts from outside to stop the combat. That report also mentions the details of the violent conflicts between Israel and Lebanon, but the only interesting information for a user who has read the previous reports is the update of the topic. With the new information added, the incident network is shown as in Figure 6, and only the bottom right part has high information value.

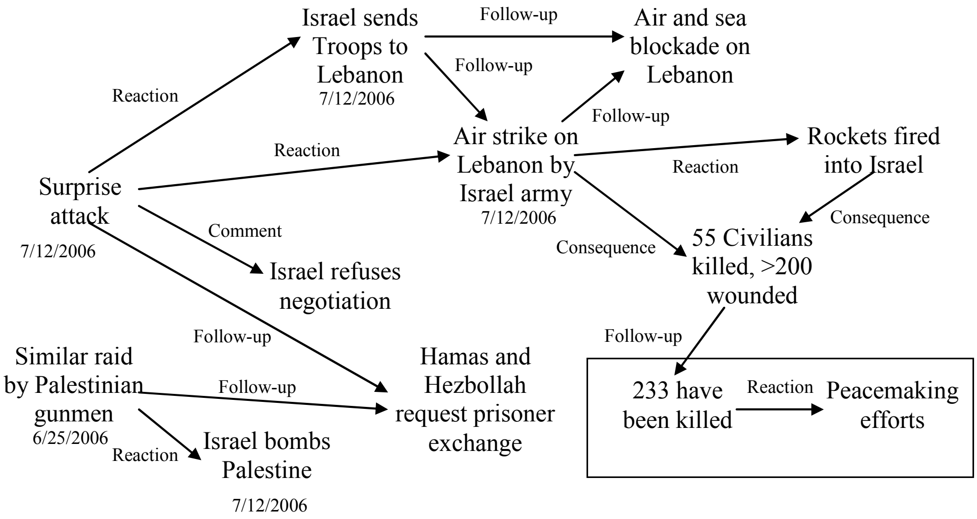

Israel sends Troops to Lebanon
7/12/2006
Follow-up
Air and sea blockade on Lebanon
Follow-up
Reaction
Follow-up
Air strike on Lebanon by Israel army
7/12/2006
Reaction
Rockets fired into Israel
Reaction
Surprise attack
7/12/2006
Comment
Consequence
Consequence
55 Civilians killed, >200 wounded
Israel refuses negotiation
Follow-up
Follow-up
Similar raid by Palestinian gunmen
6/25/2006
Follow-up
Hamas and Hezbollah request prisoner exchange
233 have been killed
Reaction
Peacemaking efforts
Reaction
Israel bombs Palestine
7/12/2006

Figure 6: Incident Network of the Israeli-Lebanon conflict

From the examples above, it can be observed that the framework of incident threading efficiently solves the redundancy problem by inserting only the new information into the incident network. Duplicate information is represented only once (although new facts will generally start with the appropriate past incidents), which saves the user much time by avoiding going through the same information repeatedly. Meanwhile, the network provides contextual information, which greatly facilitates the browsing process with links in high accuracy.

1.2 Structure and Contributions of Thesis

In the next chapter, earlier research topics related to incident threading will be described and analyzed in the circumstance of automatic news processing. The list includes, among others, topic detection and tracking, discourse analysis, novelty detection, news summarization, and information filtering. Efforts have been made to tackle the problem of news organization with success in some aspects. But with the great difficulty in this research area, many assumptions and restrictions have been made in the previous work, which simplify the task but also reduce the value of the result.

In Chapter 3, the infrastructure of incident threading is introduced in detail, which reconsiders many of the prior assumptions and builds a new framework, displaying news information in a user-friendly way. We propose an incident-based system for news organization. This framework explores the internal structure of news topics by breaking each of them into finer-grained units, and reduces information redundancy through merging content-coherent text pieces into a single incident. Work in incident threading catches the contextual information among incidents, which has heretofore not been successfully implemented with information retrieval methods. We believe that this model will ultimately improve a user's efficiency in news browsing and arouse research interest in this area.

Following the framework, several incident threading systems have been implemented. Two systems based on news stories (i.e., continuing with a previous simplifying assumption) are described in Chapters 4 and 5, and the latest passage-based implementation is in Chapter 6.

The earlier story threading work [Nallapati et al 2004] in Chapter 4 focuses on the internal structure of a news topic. Stories in the same topic form clusters, which become incidents, where the contextual information between two incidents is reduced to a binary link. We show that good pair-wise clustering and link accuracy (refer to Appendix C for evaluation measures) can be achieved with surface features and simple algorithms.

Relation-oriented story threading [Feng and Allan 2007] in Chapter 5 extends the previous model by assigning specific type labels to the links. Experiments show that a global optimization framework usually generates links of better quality, as the traditional clustering-linking method does not have the ability to correct clustering mistakes once they are made.

Chapter 6 describes passage threading, which performs news analysis at a smaller granularity. A news incident has become the aggregate of shorter text snippets that describe the same real-world happening, and the uninteresting parts (text that does not satisfy certain criteria) are ignored. Due to the lack of experimental data with judgments, we created our own data collections from news corpora and hired annotators to mark incidents in them. We design new evaluation algorithms because the existing evaluation measures do not work for this application. A baseline algorithm, which is borrowed from our previous work in Chapter 4, and a new three-stage method are described. Experiment results show significant performance improvements, on the order of 3.61% to 6.54% and 19.1% to 26.4% (on cluster-link mean and distance matrix comparison, respectively. Refer to Appendix C for details of these evaluation measures). A calibration study explores the value of the incident threading framework in real applications, and provides a performance milestone for the incident network. Surprisingly enough, an incident network that is only 25-30% accurate in a matrix comparison evaluation (see Appendix C), which has been achieved in our experiments, improves a user's ability to comprehend news quickly.

Chapter 7 summarizes the content of the thesis, lists the contributions of the work, and proposes potential research topics for the future.

The main contributions of the thesis are:

- It defines incident threading, a novel news analysis framework, on the identification of real-world occurrences in news and their contextual correlation.
- It extends prior research by exploring the internal structure of news topics (story threading).
- It introduces detailed link types to incident threading (relation-oriented story threading).
- It extends the incident threading framework from documents to passages, removing the earlier assumption that each news story discusses a single incident (passage threading).
- It displays significant performance improvements over an early implementation in both story and passage-level experiments.
- It demonstrates over 200% increase of link quality (F-value of binary links, see Appendix C) with successful application of global optimization on story pairs (relation-oriented story threading).
- It provides evidence through a calibration study that incident networks with current performance level are useful by improving efficiency of reading comprehension (passage threading).

CHAPTER 2

PREVIOUS WORK

The idea of incident threading is motivated mostly by *Topic Detection and Tracking* (TDT) and discourse analysis. TDT monitors a news stream and places the text pieces into individual topics, where each topic includes all the news events that are closely related. In addition to the effort of automatic news organization, *discourse analysis* studies the information flow in a press article. To some extent, discourse analysis is the parallel work of incident threading in another area, but the vast involvement of human beings limits its application to large corpora. There are other news processing tasks, e.g., novelty detection, news summarization and information filtering, which also aim at helping users in their browsing. In this chapter, we provide an overview of these areas and show why they have not provided an ideal framework for news organization. Overall, incident threading is related to the previous work, but it also extends the range and becomes a more realistic task. We believe that research in this direction will bring deeper understanding into automatic news analysis.

2.1 TDT

TDT [Allan 2002a] is a research program that focuses on event-based news organization. It breaks an incoming news stream into a list of topics, and each topic is "a set of news stories that are strongly related by some seminal real-world event." [Allan 2002b] As it involves the subjective understanding of news, which may differ by person, great difficulty is expected when the process is replicated in every detail. To simplify the problem, several assumptions are made in TDT:

- Topics do not overlap, which means that each news story belongs to at most one topic. This assumption is not always true since one story may mention multiple topics that are closely related to each other, but it is a reasonable simplification of the problem.
- Topics are independent. A topic is a complete entity and any possible relation with other topics is ignored. It facilitates the design of the theoretical framework, but is not true in reality. For example, TDT-3 topic "Hurricane Mitch" describes a natural disaster in 1998 that affected the Caribbean area. Seven years later, another hurricane – Hurricane Katrina – attacked the continental US. They are separate topics according to the definition in TDT, but people tend to list them together because both are similar natural disasters that happened in the same area.
- The internal structure of a topic is not important. All the evaluation metrics are topic-based, and systems invest most of their effort into assigning stories to the correct topic. Although it helps to formulate the task, the assumption is missing important information. From its definition, a topic is composed of a seminal event and other related events, so the analysis of their relations helps us to understand the topic better. However, the TDT evaluation approach intentionally ignores that and does not go deeper into the topics.

TDT had an annual evaluation from 1997 to 2004, where results from different institutions/companies were compared to decide the most effective algorithm in news organization. These are the five main tasks in a TDT evaluation:

1. Story segmentation: In the continuous stream of news text, find the story boundaries (only applicable to audio sources).

2. First story detection: Decide if an incoming story starts a new topic, or continues an existing one.
3. Cluster detection: Cluster the news stream into groups, where each group is a topic.
4. Tracking: Given a few training stories from a topic, find all the later stories that belong to the same topic.
5. Story link detection: Decide if two news stories belong to the same topic.

Starting from the pilot study in 1997 [Allan et al 1998], there were a total of eight evaluations up to TDT-2004 [Fiscus and Wheatley 2004]. A large number of models and algorithms were tried in these evaluations, and many proved quite successful [Leek et al 2002, Yang et al 2002, Yamron et al 2002, Dharanispragada et al 2002, Eichmann and Srinivasan 2002, Levow and Oard 2002, Allan et al 2002, Schultz and Liberman 2002, Chen and Ku 2002]. From the evaluation results, a topic-based news organization can be achieved at good accuracy.

To evaluate a TDT system, its output must be compared to the truth data and its performance is decided by how well they match. Some annotators were hired to mark the stories on a per-topic basis [Cieri et al 2002]. In theory, each story in the collection must be read by annotators and marked to which topic it belongs, but this method is very labor intensive. Search-guided annotation is used so that only a limited number of stories have to be annotated, and others are considered off-topic [Strassel and Glenn 2004].

Guided search usually returns stories that are very likely to be relevant, but many on-topic stories may still be missing. Since only a small portion of stories are labeled for each topic, co-training can be used to find more relevant ones from the unlabeled data. It trains weak predictors from both labeled and unlabeled data, and bootstrapping between them with different features has the potential to improve performance on classification [Blum and Mitchell 1998].

The performance is mainly measured by two different error rates, P_{miss} and P_{fa} [Fiscus and Doddington 2002]. Suppose that the user annotation returns a label target/non-target for each story-topic pair, and a TDT system also generates a yes/no judgment for the same pair. The contingency table of the user annotation/system judgment is shown in Table 1.

	Truth - target	Truth - non-target
System – yes	R^+	N^+ (false alarm)
System – no	R^- (missed detection)	N^-

Table 1: Contingency Table of User/System Judgments

The error rates are calculated as:

$$P_{miss} = R^- /(R^+ + R^-)$$
$$P_{fa} = N^+ /(N^+ + N^-)$$

These two error rates are usually inversely related, that is, when one falls, the other would rise. To determine an overall system evaluation, they are linearly combined to produce the total detection cost:

$$C_{\text{det}} = C_{miss} * P_{miss} * P_{topic} + C_{fa} * P_{fa} * (1 - P_{topic})$$

Here C_{miss} and C_{fa} are the costs of a missed detection and a false alarm, respectively, which are defined by the specific task. P_{topic} is the probability of hitting an on-topic story by randomly selecting a document from the whole collection. It is usually assigned an empirical value from previous data collections, but the actual probability differs greatly by Topic [Manmatha et al 2002].

TDT has become quite mature after eight annual evaluations. The concept *topic* has been empirically defined with detailed instructions, and building topics from a continuous news stream has achieved reasonably high accuracy [Fiscus and Wheatley 2004]. However, we still do not have a clear view how a news topic is formed, and the independence assumption of topics is often challenged. Assume that there was an explosion in Oklahoma City one day, and in a few days another bomb injured one civilian in Florida. Later the same terrorist group claimed responsibility for both incidents. Are these explosions independent or not? According to the definition, they are classified as two different topics, but the connection between them cannot be ignored.

After a few rounds of TDT evaluations, it has become clear that topic-based organization is not enough for in-depth news analysis. The concept of a topic is unclear mainly because of the granularity issue, since there can be valid topics with different sizes. In TDT 2004, the traditional topic detection task was replaced by a hierarchical structure [TDT2004], and a Hierarchical Agglomerative Clustering (HAC) algorithm with sampling [Trieschnigg and Kraaij 2005] achieved the highest performance in the evaluation [Allan et al 2003a]. Feng and Allan [2005] describe the task definition, evaluation measures and analyze the results from different participants. In comparison to the traditional flat clustering task, the hierarchical framework is more appropriate for the comprehension of topics with various granularities. From this structure, it can be observed that topics are usually composed of smaller subtopics, or incidents at a certain level, but it is still not clear how these lower-level structures are arranged to form a topic. In order to facilitate the understanding of news, we need a clearer specification of incidents and more analysis of their correlation.

2.2 Discourse Analysis

As a TDT topic is defined as a seminal event together with all related events, a natural response would be an attempt to find these individual events and indicate the relations among them. Nonetheless, if two events are randomly picked from a news topic and an annotator is asked to give the description of their relation, this simple experiment may end up with dozens of different answers after trying many people. A limited vocabulary of connection types and a detailed description (if not a definition) of each are necessary to avoid the possible confusion.

We know of no previous attempt to do this in the Information Retrieval (IR) community, but discourse analysis in journalism deals with similar problems [Brown and Yule 1983, van Dijk 1983, Schiffrin et al 2001, Gee 2005]. Discourse analysis is a general term that includes many approaches to analyzing the use of languages, and one important application of it is on news. Within the news domain, discourse analysis deals with the formation of a complete news report (mainly for news in the press), while broadcast news is usually released in shorter pieces and the context is often assumed to be available for the audience. However, models in discourse analysis may also work for broadcast news, if each piece is regarded as a part of a press article.

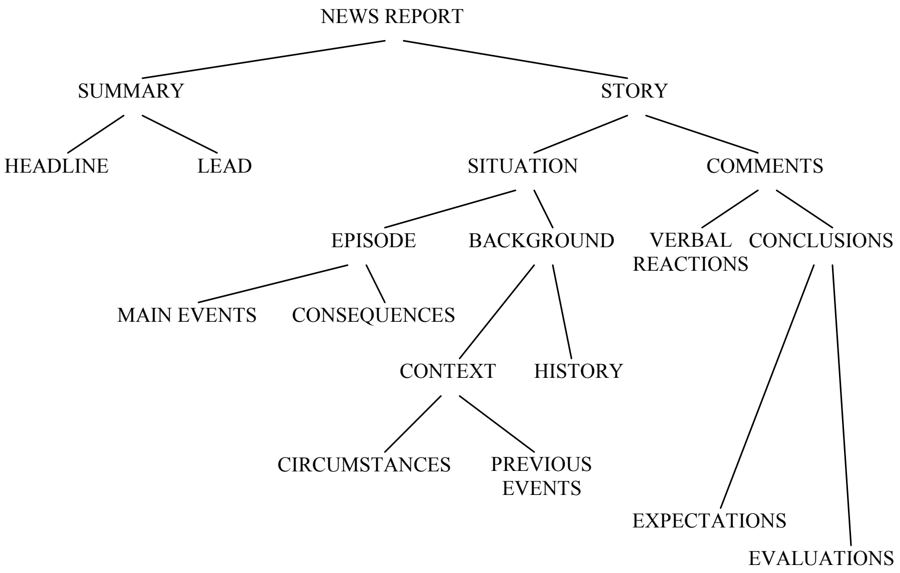


Figure 7: Hypothetical Structure of a News Schema

Figure 7 (reproduced from van Dijk [1988] page 55) shows the structure of a news schema in discourse analysis. Some of the units in the news format are of no value for our application, e.g., headline and lead, because they do not describe any contextual relation between incidents. However, others can be used to describe the organization of broadcast news events.

For the example in Figure 8, the surprise attacked conducted by Hezbollah is the MAIN EVENT of the topic, as most of the materials are directly related to it. Another raid that was carried out by Palestinian gunmen belongs to the PREVIOUS EVENTS. The retaliation of the Israeli army and further escalation of the conflict are the CONSEQUENCES of the attack, so are the damages caused by the conflict. Since there is international concern of the violence, various REACTIONS come from many parties, either involved in the conflict or not. Some of those reactions are VERBAL, and the others are real actions (CONSEQUENCES) to try stopping the violence. Although assigning each of the nodes in Figure 8 into a certain location of the structure is difficult, it is clear that many items in Figure 7 can be used in contextual news analysis.

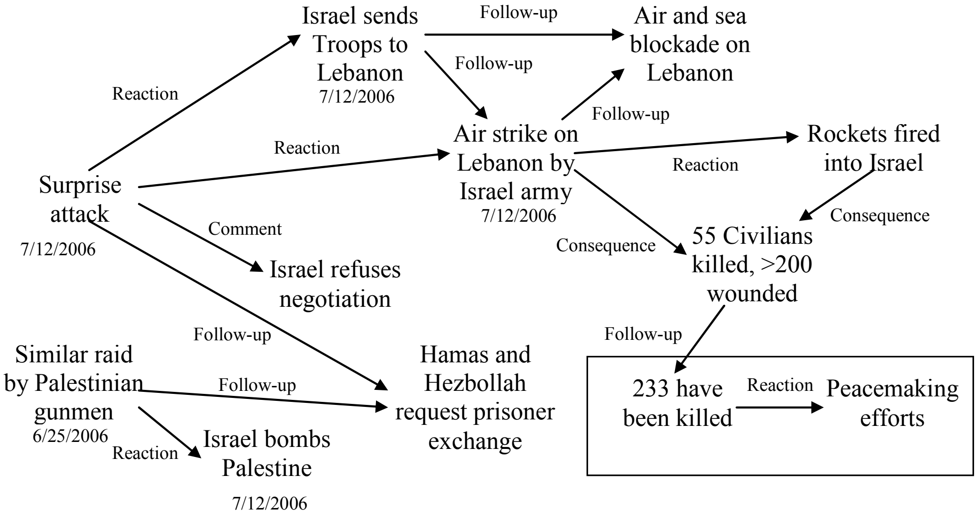

Israel sends Troops to Lebanon
7/12/2006
Follow-up
Air and sea blockade on Lebanon
Follow-up
Reaction
Follow-up
Air strike on Lebanon by Israel army
7/12/2006
Reaction
Rockets fired into Israel
Reaction
Surprise attack
7/12/2006
Comment
Consequence
Consequence
55 Civilians killed, >200 wounded
Israel refuses negotiation
Follow-up
Follow-up
Similar raid by Palestinian gunmen
6/25/2006
Follow-up
Hamas and Hezbollah request prisoner exchange
233 have been killed
Reaction
Peacemaking efforts
Reaction
Israel bombs Palestine
7/12/2006

Figure 8: Incident Network of the Israeli-Lebanon conflict (copy of Figure 6)

Van Dijk [1980] describes in detail how the micro-propositions (atomic semantic units, usually sentences or clauses) are arranged into macrostructures by the application of macro-rules. There are four main types of rules in this model – weak deletion, strong deletion, generalization and construction.

It is worth pointing out that discourse analysis methods in news, including news schemata and macro-rules, are usually applied by people instead of computer programs. In contrast, our system must be algorithm-based and avoid human intervention.

DeJong [1979] designs a system called FRUMP that automatically skims a news article and understands the main idea with details skipped. It starts with 50 "sketchy scripts" that are manually established, each for a certain type of event. Then the system integrates a parser into the natural language processing system, and makes predictions of script assignment (with 38% accuracy for news stories that match the existing scripts) along the process of reading news. It has similar abilities to discourse analysis for the study of articles that describe certain news events, but most of the content details are ignored, leaving only a summary that shows the outline. Portability is also an issue for its application to general news, because of the manual design of scripts. FRUMP is appropriate for summarization purposes, but it leaves out an important aspect of news processing – incident relation analysis, on which incident threading places the main attention.

Except for its applications in news, discourse analysis is often utilized in other domains. For example, rule generation from semantic "intersentential inference" is an important component of an Information Extraction (IE) system. The usual method of obtaining knowledge from a new domain is to manually generate heuristics throughout the reading process, but Machine Learning (ML) techniques often have the same ability of knowledge acquisition with discourse analysis at the sentence level. Wrap-Up [Soderland and Lehnert 1994] displays the main advantages of such a ML-based system – portability and scalability – in news collections. CRYSTAL [Soderland et al 1995] generalizes the acquisition process of domain knowledge by transforming the task into the identification of "concept-node definitions," which form a minimal dictionary to cover the training samples. Instead of semantic examination of separate terms, it extracts the relevance information from the sentential discourse analysis of documents.

Similarly to discourse analysis, incident threading focuses on the contextual information in news reports. Its object is a large news collection instead of a single document, and we will observe richer relation types in the new infrastructure.

Nonetheless, many concepts and terms in discourse analysis still apply to incident threading, mainly for the relation types among incidents.

2.3 Other

Besides TDT, there are other research topics that also aim at automatic processing of news to help retrieve useful information. Their frameworks and emphases are different from incident threading, but each of them is facilitating users in some aspect to obtain information they need.

It is a waste of time for a user to retrieve duplicate information, and the Text REtrieval Conference (TREC) had a novelty track that ran for three years – 2002 [Harman 2002] through 2004 [Soboroff 2004] – to address the redundancy problem. The task is designed as follows. Given a query and an ordered list of documents, the system is required to identify the relevant (matching the query) and novel (containing new information) sentences for that topic. With the novelty requirement removed, the task is exactly information filtering [Belkin and Croft 1992] at the sentence level. Topic tracking in TDT is story filtering in news, and it has achieved high accuracy on locating the on-topic stories. Unfortunately, filtering relevant sentences remains the bottleneck of the novelty track [Allan et al 2003b]. After three annual runs of the novelty track, the conclusion is that novelty detection is still a hard problem, mainly because of the limited information that each sentence contains [Soboroff 2004]. Context is often required to comprehend a sentence; and natural language understanding techniques and knowledge in cognition science are necessary to understand it better. It is very expensive to incorporate all these in an automatic system. Fortunately, features like sentence patterns can help to identify novel information [Li and Croft 2005]. In this work, we believe that novelty detection at the event level may prove more achievable, because of the longer text and additional contextual information.

News summarization [Mani and Maybury 1999] is another way of reducing the reading workload of users. Stories (or cluster of stories) are automatically abbreviated and the user only needs to read the summary, which is usually much shorter than the original text. McKenna and Liddy [1999] built a summarizer in the Tipster research program, which generates the summary of a single document or multiple documents returned from a query in DR-LINK. Later summarization systems became available on the web. Newsblaster[1] [McKeown et al 2002] clusters news stories into a hierarchical structure, where the units in the lowest level are similar to news events, and larger clusters correspond to topics. A summary of the stories in a single cluster is generated, and it provides the user with a concise version of the news information. The implementation of Newsblaster is a combination of TDT and text summarization, and it displays information in a timely manner by showing the summary of stories in a certain period. It shows the process of topic evolution by summarizing related information in certain intervals, but lacks the ability to show their intrinsic relation. NewsInEssence[2] [Radev et al 2005] is another summarization system, but one that is more user-oriented. In addition to the system-built news clusters, the user can provide a query or seed story, and a new cluster will be created based on it. NewsInEssence also provides a search function to locate interesting topics, and allows summaries at various sizes in order to

[1] http://newsblaster.cs.columbia.edu/ as of April 28, 2008.

[2] http://lada.si.umich.edu:8080/clair/nie1/nie.cgi as of April 28, 2008.

satisfy different requirements for details. In addition to those that focus mainly on news text in English, there are also summarization systems for news in other languages [Fuentes and Rodríguez 2002, Lee et al 2005] and other media [Wong 2002]. News summarization systems reduce information length and reading time, but it is still the user's responsibility to keep track of how the topic evolves over time. In the thesis, we are going to address this problem.

In some aspects, NewsInEssence is also a news filtering system, because it has the option to build a cluster summary based on a user's information need (a query or a news story in which he/she is interested). Filtering systems serve only information predicted to be interesting to users, and the key problem is to create and update the *user profile* [Pazzani and Billsus 1997]. NewsWeeder [Lang 1995] uses Minimum Description Length (MDL) to learn the profile based on the user ratings of previously read articles. It raises the percentage of interesting articles from 14% to 52% through the learning process, and also shows significant improvement over the unsupervised baseline. INFOS [Mock 1996] is a hybrid system that incorporates both keyword-based and knowledge-based algorithms. Keyword-based algorithms are generally faster, but achieve lower performance. Knowledge-based algorithms have the ability to analyze information in more detail, at the cost of higher computational complexity. A combination of them achieves high accuracy and retains scalability, but also compromises the advantage of each. Collaborative filtering [Konstan et al 1997] builds a user's profile from those of others, and greatly reduces the effort of estimating the interest of an individual user. There are many filtering systems available on the web; Kilander [1995] provides an overall comparison for eleven of them. News filtering is based on the assumption that a user is only interested in one or a few specific topic(s) and will keep a relatively stable profile. However, people are hard to predict, and such a framework requires a large amount of training data to build a precise user model, which may not be available for all applications. In our work, user profiling is not required, and the incident network is identical for everyone, while people show interest in different parts of it.

An early attempt to generate an event-level summary is temporal summarization [Allan et al 2001]. Much like the later TREC novelty task, it considers two attributes of a sentence's importance – usefulness and novelty. In a consecutive news stream, each sentence is assigned a score, which is the combination of its two attributes. Then the summary is generated by ignoring the sentences with a score below the threshold. The system shows high accuracy in the prediction of novelty, but the experiments on usefulness do not achieve obvious improvement over a simple baseline.

In this section, several research topics have been covered, which aim at reducing the workload of news readers. Some of them abbreviate information directly (news summarization, temporal summarization), and others display only a subset of the whole corpus in which the user is likely to be interested (novelty detection, filtering). Incident threading is different from them because it creates a global representation of the data (in contrast to filtering, which is user-specific), and the text is served in the original size (unlike novelty detection which removes the redundant information, duplicate descriptions are clustered together). The most important feature that distinguishes it from past work is that incident threading supplies contextual information to display the evolution of news.

CHAPTER 3

INCIDENT THREADING

To obtain a clearer view what is happening in a news stream, we believe it is necessary to go beyond the topic level, to analyze the internal structure of news topics. Ideally, the individual news events should be identified and organized into a fact network, where the edges show their relations along with the correct type. Furthermore, events in different topics may be correlated with a looser connection (the same type of occurrences, involving the same person, happening at the same place, etc.). By building such an event network, users can browse through news events, see how they appear, develop and disappear in the news stream. They can also switch from one topic to another by following the links that tie them together. Such a network is more user-friendly and efficient for the purpose of news browsing than a list of independent topics, since users tend to be interested in news in certain categories instead of a specific topic. This function is especially beneficial when a new topic appears in the news stream.

We call these news events *incidents*[1], and the graph that shows the relation among incidents an *incident network*. The process of identifying the incidents and generating the network is called *incident threading*. Whenever we talk about an event, an incident, a

[1] The concept "event" is popular in IE where it has a different meaning [Grishman and Sundheim 1996], so we replace it with *incident* to avoid confusing. An event in IE is an activity described by a sentence that involves zero or more entities. For example, "Israeli troops fought running gun battles with Palestinian civilians and security forces again today" describes an event, while "Israeli troops", "fought", "battles", "Palestinian civilians" and "security forces" are the text to extract. The event extraction tasks are usually limited to certain types of events (like conflict in the example above) and focus on the accurate identification of their arguments. Different descriptions of the same semantic content are often handled separately.

fact, an occurrence or a happening in this thesis, we refer to a specific news incident that happened sometime in the real world and/or the union of all text that describe this occurrence.

This chapter describes the framework of incident threading, including definitions of news incidents and an incident network. Main relation types that form the incident network are also introduced. At the end of the chapter, we discuss variations of an incident threading system, with the implementations described in later chapters.

3.1 Incident

Before talking about what incident threading is, or how a threading system will be implemented, it is necessary to define the basic concepts first.

1. *News story*: This is the basic unit in news distribution. Each story has a unique ID, a series of characters containing its content, and the source time which marks its time of publication. Some stories, mainly those from newswire, contain optional fields like a title, a headline or a list of keywords. Broadcast news data usually come from Automatic Speech Recognition (ASR), and those fields are often unavailable in that case. A news story usually describes one or more real-world occurrences.
2. *Main characters* (WHO): The most important named entities in the description of an occurrence that show who or what is involved in it. They can be either persons, organizations or other types of objects.
3. *Time stamps* (WHEN): Two types of time features are considered for a piece of news report. One is the publication time of the news, which is the same for

different parts of one story. The other is the absolute or relative time point (or range) that describes the time when the corresponding event happened. The latter is generally more important, but it may not be available for some reports. These should be normalized to the same format [Filatova and Hovy 2001]. Sometimes the context is required to obtain the correct time stamp, for descriptions like "on Wednesday," "the past week," "when he was 18."

4. *Location* (WHERE): The geographical position where the event happened. It is very common that the location information is not mentioned in the description, and inference from context is necessary under that condition.
5. *Action* (WHAT): The key verb that describes the actual happening of the event. Verbs often have multiple meanings (polysemy), especially for a generic term like *get* or *have*. A relatively simple method to distinguish between various meanings is to create classes of similar verbs. For example, *say*, *tell*, *criticize*, *praise* and *comment* are all verbal actions. Klavans and Kan [1998] show a successful application of classified verbs for identifying the profile of a document. In the classification experiment of violent actions described in Chapter 6, a selected list of action verbs is used to distinguish violence information from other text.

With these concepts above, what an incident is can be naturally defined.

Definition 1a: An **incident** is something that happens in the real world. It involves some main characters, occurs at a certain time or during a certain period, happens at a geographical location, and includes a specific action.

Definition 1b: An **incident** also refers to all news reports that describe the real-world occurrence, despite the vocabulary, language or medium of the report.

(a) JERUSALEM -- The Lebanese guerrilla group Hezbollah surprised Israel with a daylight assault across the border on Wednesday, leading to fighting in which two Israeli soldiers were captured and at least eight killed, and elevating recent tensions into a serious two-front battle.
(b) Israel, already waging a military operation in the Gaza Strip to free a soldier captured by Palestinian militants on June 25, immediately responded by sending armored forces into southern Lebanon for the first time in six years and holding Lebanon's government responsible for the Hezbollah assault.
(c) The toll was the highest one for the Israeli soldiers in several years, and combined with the deaths on Wednesday of more than 20 Palestinians, including many civilians, in fighting in Gaza, it was the deadliest day in the Arab-Israeli conflict since Israel withdrew from the Gaza Strip last year. Andthe violence continued into the early morning hours, when an Israeli airstrike heavily damaged the Palestinian Foreign Ministry building in Gaza.
(d) Even though Israel has military superiority in southern Lebanon and Gaza, the new fighting signaled the emergence of a conflict that has blown past the limits of local confrontation into a regional crisis.
(e) And some analysts suggested that the similarity between the Hezbollah raid and the earlier one in Gaza by fighters with the Islamic faction Hamas and its allies, both intended to gain leverage through captured Israeli soldiers, may demonstrate a growing and troubling rapport between the two groups.
(f) As with the Gaza conflict, Israel ruled out negotiations with the Lebanese captors of the Israeli soldiers. Prime Minister Ehud Olmert said he held the Lebanese government responsible for the assault by Hezbollah, the Shiite Muslim group that participates in Lebanese politics but also continues to battle Israel.
(g) "I want to make clear that the event this morning is not a terror act, but an act of a sovereign state that attacked Israel without reason," Olmert said. "The government of Lebanon, of which Hezbollah is a part, is trying to shake the stability of the region."
(h) Israel is demanding that its soldiers be returned unconditionally and that militant groups stop firing rockets at Israeli civilians from Gaza in the south and Lebanon in the north.
(i) But both Hamas and Hezbollah are holding out for an exchange for a large number of Palestinian and other Arab prisoners held by Israel.
(j) "The prisoners will not be returned except through one way -- indirect negotiations and a trade," said the leader of Hezbollah, Sheik Hassan Nasrallah, speaking to reporters in Beirut late on Wednesday.

Figure 9: Sample News Story - NYT_ENG_20060712.0202 (copy of Figure 2)

There might be some pieces of text that are meaningless for our purpose, e.g., *"Hello Bob." she said*. This sentence does not contain any useful information about occurrences, so it is not assigned to an incident. It will not be a surprise if many snippets like that are observed in a collection, but they do not provide any help in understanding the news. There is also text that includes useful information but does not describe any event happening. For example, paragraph (d) in Figure 9 analyzes the current military force contrast in Lebanon and Gaza. The paragraph helps one understand the events to which it is connected, but nothing happened at any given time because it is just general information. As the text described above does not include any occurrence, it should not be assigned to an incident.

Basically an incident is a real-world occurrence, which involves certain main characters, and happens at specific time and location. It can also be described as the union of text that contains the same (or similar) features (who, when, where, what) and describes the same thing.

3.2 Incident Network

As mentioned above, text involving exactly the same features is usually clustered into one incident. But how about two pieces of text that have some overlapping factors but differ in other aspects? For the example in Figure 10, Hezbollah performed a surprise attack into Israel, capturing two soldiers. Israel immediately retaliated by bombing many targets in Lebanon, and then we have a damage report for those conflicts. These are obviously different incidents, because the main characters do not completely match, they happen at different times, and the actions are not the same. However, from our point of view, the air strike was a response to the surprise attack, and the death and injuries are directly caused by the conflict between Israel and Lebanon. So here isolated incidents cannot provide us with the necessary information for understanding the news, like why the Israeli warplanes would strike Lebanon, and from where the deaths come. A formal structure is required to link the related incidents together.

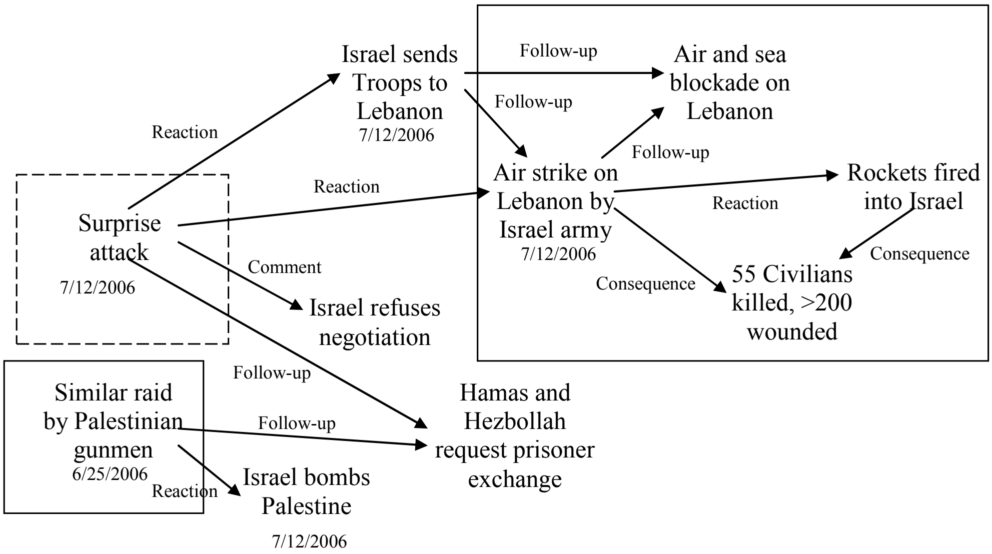

Israel sends Troops to Lebanon
7/12/2006
Follow-up
Air and sea blockade on Lebanon
Follow-up
Reaction
Follow-up
Air strike on Lebanon by Israel army
7/12/2006
Reaction
Rockets fired into Israel
Reaction
Surprise attack
7/12/2006
Consequence
Comment
Consequence
55 Civilians killed, >200 wounded
Israel refuses negotiation
Follow-up
Similar raid by Palestinian gunmen
6/25/2006
Follow-up
Hamas and Hezbollah request prisoner exchange
Reaction
Israel bombs Palestine
7/12/2006

Figure 10: Incident Network for the Stories in Figures 2 and 4 (copy of Figure 5)

Discourse analysis provides a framework that reflects the structure of a news report, and it has been shown that it can also be applied to modeling the relation of two incidents. Here some relation terms are borrowed from discourse analysis to define an incident network.

Definition 2: An **incident network** is one or more incidents connected by edges that represent certain types of contextual dependency.

Definition 3: **Incident threading** is the process of identifying incidents in news and generating an incident network.

Figure 10 shows an example of an incident network.

There are three main classes of connections in an incident network. We present them in a decreasing order of anticipated difficulty.

The first class is *logical relations*. Connections of this type specify that one incident is the necessary premise or the inevitable result of another, as judged by a normal adult's experience. They are represented by directed edges in the incident network, and each edge goes from the logical premise to the result or consequence. The following is a list of some relation types in this class:

- Prediction: If incident 1 talks about the possibility that something will or will not happen in the future, then incident 2 indicates the fact that it does or does not occur.
- Comment: If incident 1 is some real-world happening then incident 2 is someone's verbal or written feedback to it.
- Reaction: If incident 1 describes an event then incident 2 is another person's direct response (physical activities only, not including comments) to it. The subjects of the two incidents are therefore required to be different.
- Analysis: If the first incident describes some fact, then the second one shows a person's discussion of its history, reasons, details, possible results, or other related issues. The subject of the second incident is usually not directly involved in the first one.
- Background: If incident 1 is something that happened in the history or recently, and it is a probable cause for incident 2. The latter incident is usually more important. Incident 1 must be the actual description of the happening, not indirect statements or analysis made by someone as a third party.
- Consequence: If incident 1 is a real-world happening, then incident 2 is directly caused by it and describes a certain type of positive or negative result. It may be confused with the *background* relation mentioned above, but here incident 1 is the more important (main) event. The subject of incident 2 is often involved in incident 1, which distinguishes it from *reaction* where the subject is a third party.

Accurate identification of these logical relations involves great difficulty. However, experiments suggest that simple rules with term features can also achieve medium accuracy for certain types of relations (details can be found in Chapter 5).

The second class, here called *progressions*, requires weaker links than the logical relations. One incident may not necessarily lead to the other, but they often involve the same main characters, happen at a similar time and location. From the traditional TDT point of view, they discuss the same main topic and one follows the occurrence of the other. In TDT topic 20012 "Pope visits Cuba," Pope John Paul II held a mass in Santa Clara on January 22, 1998, and then visited the University of Havana on the next day. Obviously there is no logical relation between these activities (he did not have to visit the university after the mass), but they are both parts of the Pope's visit to Cuba, so an intrinsic connection between them exists. There is only one relation type in this class (named as *follow-up*), and the sequence is decided by the time order of the incidents. Links in this class are shown as directed edges, pointing from the earlier incident to the later one.

The third class is called *weak relations*. It occurs when two incidents do not have a strong logical link, but contain some common factor(s), like involving the same person, happening at the same time and/or geographical location. If there is an overlapping feature in two incidents, they will be linked by this matching factor. Dependencies in this class are represented by undirected edges in the incident network, because there is usually no priority or built-in order from the overlapping feature. The main types in this class include:

- Same character: If the main characters – e.g. persons, organizations – are the same in the two incidents.
- Same place: If two incidents happen at the same geographical location or two locations very close to each other. In the case that one location is a larger geographical domain containing the other, they should be treated as overlapped, but with a smaller confidence score.
- Concurrent: Two incidents happen at approximately the same time.
- Similar event: The incidents describe the same type of events, e.g., two different hurricanes that happen at different time and locations.

Occasionally two incidents can share more than one feature, so it is possible to see multiple weak links between an incident pair.

The first two classes are strong relations that usually exist within a news topic, and the last class mostly goes between topics and establishes a global incident network. It facilitates a user in navigation through the news, because the weak connections can lead the user from one interesting topic to another. It also helps to find all information related to certain entities in which the user shows interest.

3.3 Summary

In this chapter we have defined the framework of incident threading. An incident is an occurrence in the real world that involves certain main characters and happens at a specific time and location. It can also be used to refer to the union of all text describing the occurrence. Multiple incidents form an incident network, where links exist between related items. We define three classes of links – logical, progressional and weak relations – plus some specific types in each.

For the implementation of an incident threading system, there are two important decisions to make. The first is the basic unit of text in the system design. From our observation, a news story usually contains more semantic information, which makes it easier to understand, but sometimes a story mentions more than one occurrence. In contrast, a passage is shorter and often requires contextual information to know its content entirely, but has better semantic agreement. The second choice is on the contextual links. It is not very difficult to tell if a relation exists between two incidents, but marking their link type may be a subjective task. We can either go with binary links, which is easier to annotate and implement, or require the relation type to be marked for each link.

With different answers to these questions, there can be four combinations in the system implementation. In our previous work of story threading (Chapter 4), we select news stories as the basic semantic units and ignore link types. When type information is considered, it becomes the relation-oriented story threading work in Chapter 5. Passage threading in Chapter 6 analyzes news at a smaller granularity (passages instead of stories), and limits the range of news to a specific subject (violent actions in our experiment). Under that scenario, the vocabulary of relations is very limited, so the type is ignored. We have also tried passage-level incident analysis of general news for richer contextual relations. Unfortunately, the poor inter-annotator agreement prevents us from conducting further experiments with that setting (Appendix A).

Next we introduce the three successful implementations mentioned above, starting from the earliest story threading work.

CHAPTER 4

STORY THREADING

The earliest work in incident threading was published in 2004 [Nallapati et al 2004]. As an extension of TDT, *story threading* keeps the existing assumption that a story is the basic unit in news processing and all text in a story describes the same content. As the formation of news topics was not a concern of the TDT research program, this prior work of ours devotes its main effort to the organization of the internal structure of topics. (The framework is called *event threading* in the paper, but we will use *story threading* to describe it for better agreement with other implementations in this thesis.)

4.1 Model

As the earliest attempt to organize news at the incident level, story threading tries to capture the news incidents within a TDT topic and the organization among them. An incident is defined as “something that happens at some specific time and place,” (the definition is actually for a news event, but we treat these two terms as exchangeable [Yang et al 1999]) and incidents in the same topic are shown in a Directed Acyclic Graph (DAG). Figure 11 displays the ideal incident model of the three stories about the Israeli-Lebanon conflict in this framework. An edge from incident A to incident B means that there is some correlation (or dependency) between them, either logical (A causes B to happen) or progressional (A precedes B in time). However, the logical and progressional relations are nontrivial to distinguish, and a clear boundary is not established between them in this work.

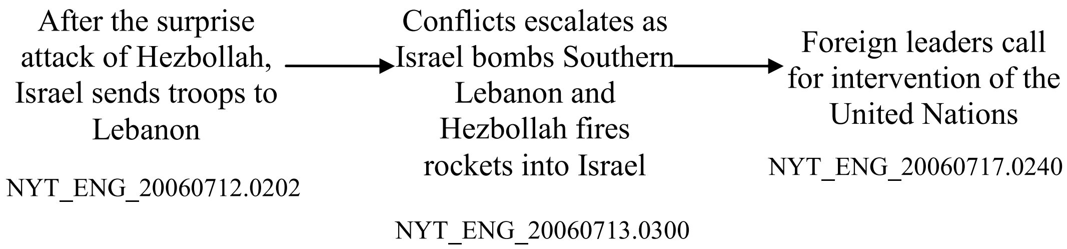

After the surprise attack of Hezbollah, Israel sends troops to Lebanon
NYT_ENG_20060712.0202
Conflicts escalates as Israel bombs Southern Lebanon and Hezbollah fires rockets into Israel
NYT_ENG_20060713.0300
Foreign leaders call for intervention of the United Nations
NYT_ENG_20060717.0240

Figure 11: Ideal Incident Model of the Israeli-Lebanon Conflict in Story Threading

The story threading framework builds on the assumptions of the TDT program, so many rules in TDT still apply. These are the main simplifications that have been made in story threading:

- Each topic is independent: With this assumption, only relevant stories from the same topic need to be compared. TDT also ignores the inter-topic relationship, but this simplification is assumed in story threading mainly to reduce the computational complexity.
- Dependencies between incidents are binary: There can be many different types of dependencies, and the assignment of such labels is subjective to some extent. To avoid confusion in this early attempt, the relation is simplified to either related (1) or non-related (0). In the next chapter we will describe a different framework that takes the link types into consideration.
- A story is the smallest news unit: This claim assumes that all the text in a news story is describing the same happening – TDT has the same simplification. From previous experience in the TDT research community, the claim is a useful assumption to reduce the amount of work in the data annotation process, as

passage-level markup is much more complex. In Chapter 6 we will see another implementation that is established on passages instead.

4.2 Algorithms

In the implementation of a story threading system, there are mainly two steps. First, all stories in the same topic are compared to each other, and similar ones are merged into a cluster. Each cluster at the end of the first step corresponds to a news incident. In the second step, two incidents with certain relationship are linked by an edge. The edge shows a “preceding” relation, as it goes from the earlier incident to the later one.

The first step is a clustering process based on agglomeration [El-Hamdouchi and Willett 1989]. There are many algorithms that can be applied to text clustering [Willett 1988]. They include Hierarchical Agglomerative Clustering (HAC), single pass, K-means and bisecting K-means [Steinbach et al 2000]. HAC forms a hierarchical structure, where clusters in different granularities can be observed at various levels, and the final configuration is often taken at a certain height of the hierarchy. HAC is stable since its result does not rely on the initial parameters, but it has at least quadratic computational complexity and is often too time-consuming when the number of data points is large. Single pass clustering is fast and easy to implement, where each incoming sample is compared to all existing clusters, but its result relies highly on the order of the data stream. Regular K-means requires a preset number of clusters, which is hard to decide for many applications. Bisecting K-means, a variant of K-means, picks one cluster to split in each round, and generates a hierarchical structure top-down (in contrast to the bottom-up process of HAC). Another option is multi-way clustering [Bekkerman et al 2005] that simultaneously clusters documents, words and other features. It has been observed to achieve better performance than “one-way” algorithms (clustering for documents only). As the number of documents in a topic is usually small, we use HAC with a threshold to halt the process.

At the beginning of the clustering step, each on-topic story is converted into a tf·idf[1] term vector by the Lemur toolkit[2]. For a story *s*, the i-th element of the corresponding vector *u* is

$$u_i = tf_i \cdot idf_i$$

where the tf component is the base form, i.e., the number of occurrences of the i-th term in story *s*. We use the Inquery idf formula [Callan et al 1992]:

$$idf_i = \log \frac{n+1}{df_i + 0.5}$$

where n is the number of stories in the collection, and df_i is the number of stories in which the i-th term appears.

Then a cosine similarity matrix is calculated for all the story pairs, where the similarity of two term vectors *u* and *v* is

$$\cos(u,v) = \frac{u \bullet v}{\|u\| \cdot \|v\|}$$

The enumerator is the inner product of two vectors, and the denominator is the product of the lengths of *u* and *v*.

[1] A widely-used form of Vector Space Model in IR [Sparck Jones 1972].

[2] Joint work of Carnegie Mellon University and University of Massachusetts Amherst. http://www.lemurproject.org/ as of April 28, 2008.

The agglomeration process starts with each story as a cluster, and the most similar cluster pair is merged in every round, until the maximal similarity is below the preset threshold. When calculating the similarity of two clusters each consisting of one or more stories, there are three commonly-used methods:

1. Average link: the average of all pair-wise similarities of documents across the cluster boundary.
2. Complete link: the minimum of all similarities. In another word, all the story pairs must be similar enough for the clusters to be claimed similar.
3. Single link: the maximum of all similarities. Contrary to complete link, a single similar story pair is enough to declare the clusters as similar.

In the preliminary experiment, which is conducted in a small subset of the training corpus, average link consistently outperforms the other two. Therefore, we believe that it is the preferred method in this setting and apply average link in all later experiments.

In addition to the term vectors, other features like person names, locations and time stamps are also applied. Person names and locations are indexed as full phrases, and the similarity based on these features is determined by whether two stories share the same person name or location. Then the similarities from these features are merged with the cosine similarity through weighted sum:

$$wsum(s_1, s_2) = w_1 \cos(s_1, s_2) + w_2 Loc(s_1, s_2) + w_3 Per(s_1, s_2)$$

Here w_1, w_2, w_3 are the weights assigned to the individual features. $Loc(\cdot)$ is 1 if there is at least one location that appears in both stories, otherwise it is 0. $Per(\cdot)$ is defined similarly.

Application of time stamps improves the performance through time decay that is described below. Although overlap of named entities or geographical locations is a strong indication of the same occurrence, the other features do not show the benefit that we have expected. The techniques used in the first step are:

- *Baseline*: Agglomerative clustering based on cosine similarity of term vectors.
- *Time decay*: According to the time difference between two news stories, a decay factor is multiplied to the cosine similarity used in the baseline:

  $$TD\_sim(u,v) = sim(u,v) \cdot e^{-\frac{|t_1 - t_2|}{T}}$$

  Here t_1 and t_2 are the time stamps of the two stories, and T is the duration of the whole topic (the time difference between the earliest and the latest stories).
- *N(T)*: The number of truth incidents is used to determine when to stop the agglomeration process. This feature is not always available, so using it makes the experiment seem like cheating. Nevertheless, it does not have any observable influence on the performance (see Table 2).

There are three algorithms in the first step. The first is the baseline, in which only term vectors are used for similarity calculation. The second method (baseline + time decay) modifies the similarity with a decay factor based on the time difference between two stories. The third algorithm (baseline + time decay + N(T))uses the same similarities as the second, but halts the agglomeration process when the number of clusters equals the number of incidents in that topic.

The second step goes through all clusters created in the previous stage, and creates links between appropriate pairs. As incident dependency has not been explicitly modeled before, we start with two simple features. First, the similarity of two clusters shows how close their content is, and it is an indication of their correlation. Second, each story comes with a publication date, and a cluster can be assigned a time stamp based on the earliest story in it. These stamps are utilized to create an order for the clusters (incidents). These are the algorithms used in the second step:

- *Baseline*: This approach assumes dependency on all incident pairs (fully linked graph).
- *Nearest parent*: The only parent candidate for an incident is its preceding incident, and the link is formed when their similarity is higher than the threshold.
- *Threshold*: All edges with a similarity score (between the two incidents connected by the edge) over the threshold are kept.
- *MST*: This algorithm starts from the strongest link and generates a Maximum Spanning Tree (MST) with a greedy algorithm [van Rijsbergen 1979].
- *Best similarity*: At most one parent for each node can be selected, which has the highest similarity.

4.3 Evaluation

For the data collection in the experiment, we selected 28 topics from TDT-2 and 25 topics from TDT-3[1]. Each of these topics contains at least 15 stories from CNN headline news. For a topic with more than 30 CNN stories, only the first 30 were kept. Then we hired an annotator to separate the documents in each topic into individual incidents and mark their relation if one exists. To ensure quality, we annotated three other topics on our own and asked the annotator to work on them first. For any disagreement between our results and the annotator's, we had in-depth discussions to make sure that we could understand the news information in the same way.

In story threading, a system is evaluated on the accuracy of both clustering (merging stories into news incidents) and dependency (identifying the relation between two incidents). For a randomly selected story pair s_i and s_j, the precision for clustering (CP) is defined as:

CP = P(s_i, s_j in the same truth incident | s_i, s_j in the same system cluster)

And the clustering recall (CR) is formulated the other way.

CR = P(s_i, s_j in the same system cluster | s_i, s_j in the same truth incident)

The pair-wise evaluation is an innovative method for clustering algorithms. The traditional measure in TDT is to find the most similar cluster in the ground truth for a system-generated cluster, and then calculate how similar they are. Because of granularity issues, it is common to see one-on-multiple and multiple-on-multiple mapping. The pair-wise method effectively solves this issue by avoiding the consideration of whole clusters.

However, these evaluations have a clear magnifying effect on errors. Suppose that a system outputs a cluster that is identical to the ground truth, we have 100% CP and 100% CR. When half of the cluster is removed, CP is still 100%, but CR becomes 25% instead of 50%. Therefore, we expect smaller evaluation numbers when using the pair-wise comparison.

The dependency accuracy is defined similarly:

DP = P(incident(s_i) $\rightarrow$ incident(s_j) | cluster(s_i) $\rightarrow$ cluster(s_j) in system)

DR = P(cluster(s_i) $\rightarrow$ cluster(s_j) in system | incident(s_i) $\rightarrow$ incident(s_j))

[1] Both corpora are available from the Linguistic Data Consortium (LDC), with catalog

incident(s_i) is the incident to which story s_i belongs, cluster(s_i) is the cluster to which story s_i is assigned, and an arrow means that there is a link going from the incident (cluster) preceding the arrow to the one after it.

4.4 Experiments

We designed experiments to verify that the incident model can be established with simple algorithms. The 53 annotated topics (28 from TDT-2 and 25 from TDT-3) are randomly divided into two parts, where 26 are in the training set and 27 in the test set.

The evaluation results (on the test set) of the individual clustering and dependency steps are listed in Tables 2 and 3, respectively. In order to show the performance of the dependency algorithms only, the second experiment was run on the annotated incidents instead of the noisy output of the previous step. A system marked with an asterisk shows significant improvement on the F-value[1] (CF or DF) over the baseline in a one-tailed t-test[2].

In Table 2, the application of time decay improves both precision and recall. This observation proves our assumption that stories in the same incident tend to be close in publication time. We have also expected that the introduction of N(T) (using the actual number of incidents as the stopping criterion instead of a threshold) will halt the agglomeration process at the correct granularity, but the experiment does not show apparent benefits in comparison to the baseline + time decay algorithm.

number LDC2001T58 for the text collection.

[1] Harmonic mean of precision and recall. It is a commonly-used measure for the evaluation of information retrieval or extraction systems, as the harmonic mean calculation requires appropriate trade-off between precision and recall of a system.

[2] http://en.wikipedia.org/wiki/T_test as of April 28, 2008.

Model	CP	CR	CF=2*CP*CR/(CP+CR)	P-value in t-test
Baseline	0.44	0.67	0.50	-
Baseline+ time decay	0.48	0.70	0.54	0.014*
Baseline+ time decay + N(T)	0.43	0.76	0.54	0.025*

Table 2: Comparison of Clustering Algorithms

Model	DP	DR	DF=2*DP*DR/(DP+DR)	P-value in t-test
Baseline	0.50	0.94	0.63	-
Nearest parent	0.61	0.60	0.60	-
Threshold	0.57	0.75	0.64	0.24
MST	0.70	0.68	0.69	0.22
Best similarity	0.71	0.74	0.72	0.04*

Table 3: Comparison of Dependency Algorithms (based on perfect clustering)

Most algorithms listed in Table 3 achieve improvement over the baseline (except nearest parent), and *best similarity* is the only one that shows significant difference. We expect that the combination of the best algorithms in the individual steps would yield the highest performance, but the experiments prove otherwise.

Table 4 shows the performance (on the test set) when the clustering and dependency algorithms are merged to generate an overall evaluation score. Although TD (time decay) greatly improves the clustering performance, it also unexpectedly lowers the dependency accuracy, which brings the joint F-value for most algorithms below the baseline (since harmonic mean is affected more by the smaller value, which is DF in this case). The only method with a joint F-value higher than the baseline is *Cos+TD+Simple threshold*, which will be the baseline of later experiments (Chapters 5 and 6). The clustering step remains the bottleneck of performance for the whole system, as errors in it are magnified in the next step.

Model	CF	DF	JF=2*CF*DF/(CF+DF)	P-value
Baseline(cos+complete-link)	0.36	0.43	0.39	-
Cos+TD+Nearest parent	0.50	0.21	0.30	-
Cos+TD+Best similarity	0.54	0.26	0.35	-
Cos+TD+MST	0.54	0.28	0.37	-
Cos+TD+Simple threshold	0.44	0.42	0.43	0.0081*

Table 4: Performance on the Combined Run of Clustering and Dependency

From the experiment results, moderate accuracy can be achieved in story threading with simple algorithms and easy-to-extract features, but the assumptions in this model leave ample space for further development.

In the next chapter, we introduce an extension to story threading, which expands the binary relation with type labels and provides a novel view of the incident network.

CHAPTER 5

RELATION-ORIENTED STORY THREADING

In the previous chapter, two incidents are either related or not related, but the actual relation has been ignored. However, from a reader's point of view, any "related" incidents are linked by a certain *type* of contextual connection. Next we describe an enhanced framework for story threading, where the focus is the accurate identification of the relation type between two incidents or two stories [Feng and Allan 2007].

5.1 Correlation Rules

In order to understand better how human beings establish connections between incidents, the example in Figure 12 can be used to describe this recognition process. In the Israeli-Lebanon conflict topic, the seminal event is the surprise attack of Hezbollah. Then for the statement of the Israeli government, the report says that Israel refuses negotiation with the Lebanese captors of the Israeli soldiers. When someone reads the text "Lebanese captors," he/she will go back to previous passages to discover who these captors are. Under normal conditions, the description of the surprise attack should be found and matched to the current paragraph. Then we know that Israel is talking about Hezbollah, and the government is not going to make any exchange with the Lebanese militant for the return of its soldiers. Here a *comment* link is created with a small amount of reasoning.

(a) JERUSALEM -- The Lebanese guerrilla group Hezbollah surprised Israel with a daylight assault across the border on Wednesday, leading to fighting in which two Israeli soldiers were captured and at least eight killed, and elevating recent tensions into a serious two-front battle.

(b) Israel, already waging a military operation in the Gaza Strip to free a soldier captured by Palestinian militants on June 25, immediately responded by sending armored forces into southern Lebanon for the first time in six years and holding Lebanon's government responsible for the Hezbollah assault.

(c) The toll was the highest one for the Israeli soldiers in several years, and combined with the deaths on Wednesday of more than 20 Palestinians, including many civilians, in fighting in Gaza, it was the deadliest day in the Arab-Israeli conflict since Israel withdrew from the Gaza Strip last year. Andthe violence continued into the early morning hours, when an Israeli airstrike heavily damaged the Palestinian Foreign Ministry building in Gaza.

(d) Even though Israel has military superiority in southern Lebanon and Gaza, the new fighting signaled the emergence of a conflict that has blown past the limits of local confrontation into a regional crisis.

(e) And some analysts suggested that the similarity between the Hezbollah raid and the earlier one in Gaza by fighters with the Islamic faction Hamas and its allies, both intended to gain leverage through captured Israeli soldiers, may demonstrate a growing and troubling rapport between the two groups.

(f) As with the Gaza conflict, Israel ruled out negotiations with the Lebanese captors of the Israeli soldiers. Prime Minister Ehud Olmert said he held the Lebanese government responsible for the assault by Hezbollah, the Shiite Muslim group that participates in Lebanese politics but also continues to battle Israel.

(g) "I want to make clear that the event this morning is not a terror act, but an act of a sovereign state that attacked Israel without reason," Olmert said. "The government of Lebanon, of which Hezbollah is a part, is trying to shake the stability of the region."

(h) Israel is demanding that its soldiers be returned unconditionally and that militant groups stop firing rockets at Israeli civilians from Gaza in the south and Lebanon in the north.

(i) But both Hamas and Hezbollah are holding out for an exchange for a large number of Palestinian and other Arab prisoners held by Israel.

(j) "The prisoners will not be returned except through one way -- indirect negotiations and a trade," said the leader of Hezbollah, Sheik Hassan Nasrallah, speaking to reporters in Beirut late on Wednesday.

Figure 12: Sample News Story - NYT_ENG_20060712.0202 (copy of Figure 2)

Although it seems straightforward to a normal person, designing a computer program with the same capability is difficult. It requires abilities in natural language understanding and artificial intelligence that are still beyond state-of-the-art research.

However, it can be observed from the example that certain relations among incidents often exist in analogous scenarios. For example, legal cases usually involve a crime, an investigation, zero or more suspects, arrests, a trial, a verdict and a sentence. Furthermore, relations among these parts are usually fixed. Schank and Abelson [1977] find similar phenomena in the understanding of human knowledge, and they create scripts for scenarios in real life (e.g., restaurant script[1]). Here the term "script" is borrowed from their work and one script is generated for each circumstance, which includes a list of rules for possible link types under that scenario. Rules in the broad area of science and discovery news are shown in Table 5 [Feng and Allan 2007].

Index	Incident 1	Incident 2	Requirement	Link Type
1	Prediction	General or Damage	Similarity	Prediction
2	General	Damage	Similarity	Consequence
3	General or Damage	Comment	Similarity	Comment
4	Preparation	General	Similarity	Preparation
5	General	General	Similarity and Time order	Follow-up

Table 5: Correlation Rules in Science/Discovery News

The fields in each row are rule index, the type to which incident 1 belongs, the type to which incident 2 belongs, the requirement(s) for the two incidents to form a link, and the type of link, respectively. These are some sample incidents in the various types shown in Table 5:

Prediction: there will a meteor shower.

Damage: the meteor shower did not cause damage to satellites as expected.

Comment: scientists highly compliment the finding of a dinosaur nesting site.

Preparation: earlier work before launching the Zarya spaceship.

General: Octuplets born in Houston.

These rules are usually generated by human beings from the observation in a news corpus. As intensive manual work is involved in the process, this approach is

[1] The main steps in the restaurant script include: customer enters restaurant, customer finds seat, customer sits down, waiter/waitress gets menu, etc.

inappropriate for an application in a large dataset. An alternative method is to run ML methods on annotated data to create rules automatically [Langley and Simon 1995]. Due to the limitation of ML algorithms, a large amount of data is often necessary to identify clear patterns for automatic rule induction. Unfortunately, we do not have a large text collection that is fully annotated with contextual information, so we settle with manually-generated rules within this chapter.

After we define the link types, contextual information can be represented more accurately in an incident network. The network composed of the three stories in the Israeli-Lebanon conflict is shown in Figure 13.

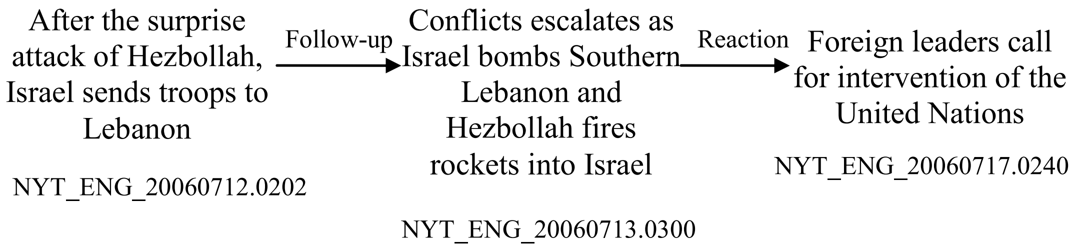

After the surprise attack of Hezbollah, Israel sends troops to Lebanon
NYT_ENG_20060712.0202
Follow-up
Conflicts escalates as Israel bombs Southern Lebanon and Hezbollah fires rockets into Israel
NYT_ENG_20060713.0300
Reaction
Foreign leaders call for intervention of the United Nations
NYT_ENG_20060717.0240

Figure 13: Ideal Incident Model of the Israeli-Lebanon Conflict in Relation-Oriented Story Threading

5.2 Improved Story Threading

With the correlation rules in Table 5, it is feasible to establish a link between related incidents with the corresponding type. In this section, we introduce a two-stage algorithm that builds on story threading (Chapter 4) but includes more features and generates links with rules instead of cluster similarity. For comparison purposes, a method in story threading becomes the baseline, with a slightly different implementation.

Model	CF	DF	JF=2*CF*DF/(CF+DF)	P-value in t-test
Baseline(cos+complete-link)	0.36	0.43	0.39	-
Cos+TD+Nearest parent	0.50	0.21	0.30	-
Cos+TD+Best similarity	0.54	0.26	0.35	-
Cos+TD+MST	0.54	0.28	0.37	-
Cos+TD+Simple threshold	0.44	0.42	0.43	0.0081*

Table 6: Performance on the Combined Run of Clustering and Dependency (copy of Table 4)

From Table 6, the best-performing algorithm in story threading [Nallapati et al 2004] is Cos+TD+Simple threshold. Since time decay requires the duration of each TDT topic, which is unavailable without topic information, it is ignored in this implementation. Like the algorithm in story threading, the similarity of two term vectors is the cosine of the angle between them:

$$\cos(u,v) = \frac{u \bullet v}{\| u \| \cdot \| v \|} \qquad (1)$$

The numerator is the inner product of the tf·idf vectors of the corresponding stories, and the denominator is the product of their lengths.

The tf component is a variation of the one used in Okapi [Robertson et al 1998]:

$$\hat{tf_i} = \frac{tf_i}{tf_i + 0.5 + 1.5 \times \frac{len_p}{avg(len_p)}} \qquad (2)$$

In the equation above, tf_i is the frequency of the i-th term in story p, len_p is the length of p (number of terms in it), and $avg(len_p)$ is the average story length in the whole collection.

For the idf component, we use the Inquery normalized idf [Callan et al 1992]:

$$idf_i = \frac{\log \frac{n+0.5}{df_i}}{\log(n+1.0)} \qquad (3)$$

Here df_i is the document frequency of the i-th term, and n is the number of stories in the collection. Then each element in the term vector is the product of the corresponding tf and idf components:

$$u_i = \hat{tf_i} \cdot idf_i$$

There are many different formats of tf·idf with slight changes. In comparison to the simpler equations in Section 4.2, this version achieves higher performance with its additional normalization and parameter tuning [Connell et al 2004], so we select it in this implementation with the assumption that it will work as well in the different framework.

For the baseline method, we perform an agglomerative process as in story threading, with similarities calculated by Equation (1). Starting from individual stories, the most similar cluster pair is merged in each round, until the maximal similarity is below a preset threshold. After the agglomeration stops, all incident (corresponding to a cluster after the previous step) pairs are compared. If the similarity between two incidents is over the link threshold, a link is created between them, which points from the earlier incident to the later one. There is no type associated with the link, same as the implementation in Section 4.2. This process is identical to the one in Chapter 4, except that time decay is not applied and tf·idf takes a different formula.

The two-stage algorithm also starts with an agglomeration process, but the similarity calculation is modified to include more news-specific features. In addition to the full text, there are other useful features in the content of a news story, including main characters, locations, time stamps and key verbs. These features are extracted by an Automatic Content Extraction (ACE)[1] system from New York University[1].

```
<DOC>
<DOCNO>CNN19981101.1600.0488</DOCNO>
<TIME>
1966   1998-11   1999
</TIME>
<LOCATION>
their peak   Denver
</LOCATION>
<EVENT>
racing   hit
</EVENT>
<ENTITY>
a shield   Scientists   researchers   Leonid
</ENTITY>
<TEXT>
Scientists are racing to develop a shield to protect satellites from
a Severe meteor storm. The Leonid meteor storm happens every year
in mid-November. But this year and next year are expected to be more
intense than usual. That could put telecommunications satellites at
risk, so researchers at the university of Denver are developing a
shield of aluminum foam. The Leonid showers last hit their peak in
1966, but very few satellites were in orbit then.
</TEXT>
</DOC>
```

Figure 14: Sample Story in Two-stage Algorithm

Figure 14 shows a short sample story with the extracted features (<ENTITY> for main characters, <LOCATION> for geographical locations, <TIME> for time stamps and <EVENT> for key verbs). These features are also indexed as plain text in individual fields of the snippet, and can be viewed as various representations of the same description [Ogilvie and Callan 2003]. With all these features, the cosine similarity (Equation 1) of corresponding fields can be calculated between a story pair p_i and p_j, and

[1] http://projects.ldc.upenn.edu/ace/ as of April 28, 2008.

the overall similarity of this pair is the weighted sum of similarities based on various features.

$$sim(p_i, p_j) = \sum_{k=1}^{l} w_k sim_k(p_i, p_j)$$

Here $sim_k(p_i, p_j)$ is the similarity of p_i and p_j based on the k-th feature, and w_k is the weight associated with it. The weights satisfy $\sum_{k=1}^{l} w_k = 1$ where $l = 5$ in our experiment.

These weights are empirically adjusted for the data to achieve the best performance (by tuning weight assignment among features), but the first feature (term vector of the full text) usually receives the highest importance. In our experiments, the weight assignment is

$$(w_{term}, w_{who}, w_{where}, w_{when}, w_{what}) = (0.8, \quad 0.1, \quad 0.1, \quad 0, \quad 0)$$

After the incidents are formed with the agglomeration process, we apply correlation rules in Table 5 to establish links between appropriate pairs. Since these rules require the type information of each incident, a classifier (e.g., BoosTexter [Schapire and Singer 2000]) is necessary to assign a label to each. First, we collect some news stories and mark them with the category of information they describe (prediction, comment, damage, preparation, general or background). Then the annotated data are supplied to the classifier to train the model. Next, all other news stories are classified into one of the categories above. The type of an incident is determined by the distribution of labels in it, through majority voting. Finally, we compare each incident pair to the threading rules in Table 5, and create a link if one rule applies.

[1] Proteus, http://nlp.cs.nyu.edu/index.shtml as of Apr 28, 2008 [Grishman and Hirschman

Science/discovery News	
Topics	6
Number of stories	280
Labels	Prediction, Comment, Damage, Preparation, General, Background (no occurrence information, story ignored)
Number of stories in each class	37, 6, 2, 21, 209, 5

Table 7: Statistics of the Classification Experiment

Table 7 shows some statistics of the classification experiment. In a 3-fold cross validation (train the classifier with 2/3 of the stories, then classify the other 1/3), BoosTexter returns a 17% error rate for all stories.

5.3 Global Optimization

If two stories are randomly selected from a collection, there are two different forces that interfere with each other and try to determine their relationship. One force is the similarity between the stories, which (if high enough) tends to pull them together and merge them into the same incident. The other is their satisfaction of a correlation rule that attempts to push them apart and place them into two different (but connected) incidents. These options are mutually exclusive, and each has some probability (or score) associated with it. For a story pair, their relation can be in one of three possible states. They are either in the same incident, connected by some relation, or not related at all. These relations can be encoded as -1 (not related), 0 (in the same incident) or a positive integer (connected, with the value showing the link type).

When the pair-wise competition is expanded to the whole collection, it becomes a global optimization problem. With a collection of *n* stories, an *n*n* relation matrix *R* can

1986].

be established. Note that links are directional, so R_{ij} and R_{ji} are encoded differently if they are positive (which means that the story pair is linked by a certain relation). When there is a link of type r (refer to Table 5, r=1 means a prediction link) going from incident i to j, R_{ij}=2r-1, and R_{ji}=2r. In thermodynamics, a state with lower energy is usually stabler. Likewise, for a global score function defined on this relation matrix, a larger score means a more appropriate news organization.

$$S = \sum_{1\le i,j\le n}^{i\neq j} score(i, j, R_{ij}) \qquad (4)$$

In this equation,

$$score(i, j, R_{ij}) = \begin{cases} c & R_{ij} = -1 \\ Sim(p_i, p_j) & R_{ij} = 0 \\ Rule(p_{i,}p_j, R_{ij}) & R_{ij} > 0 \end{cases}$$

Here c is a small constant assigned to unrelated snippet pairs (0.09 returns better results in our experiment). $Sim(p_i, p_j)$ is the similarity of the two stories. $Rule(p_{i,}p_j, R_{ij})$ is a function that tells how well the pair fits the rule of relation R_{ij}, and its format differs by rule. For most rules in Table 5,

$$Rule(p_i, p_j, R_{ij}) = sat(p_i \doteq R_{ij}^{I_1}, p_j \doteq R_{ij}^{I_2}) \times \min(Sim(p_i, p_j), R_{ij}^{cap})$$

where $sat(\cdot)$ is a function that assigns a weight according to the number of true predicates, $\doteq$ means that the story is in the same type as the incident in the rule, and R_{ij}^{cap} is an upper bound of the similarity (we set it as 0.05 from empirical data). The last rule in Table 5 (follow-up) also requires the time stamps of the stories to be in the right order.

It is worth pointing out that the design of this function requires a rich relation that can be effectively distinguished from other alternatives, as the formula often contains heuristic information from our observations.

The relation matrix *R* has n^2 parameters, but they are not independent of each other. The relation between two stories must be symmetrical (note that R_{ij} and R_{ji} are not equivalent when there is a link between *i* and *j*). Furthermore, when two stories are in the same incident, they must have the same relation to all other entities. The restrictions for the global optimization problem are,

$$\begin{aligned}
&\forall i, j \quad R_{ij} \leq 0 \quad \Rightarrow R_{ji} = R_{ij} \\
&\forall i, j \quad R_{ij} > 0 \wedge R_{ij} \equiv 1(\bmod 2) \quad \Rightarrow R_{ji} = R_{ij} + 1 \\
&\forall i, j, k \quad R_{ij} = 0 \quad \Rightarrow R_{ik} = R_{jk}
\end{aligned}$$

With the restrictions above, there is no explicit solution for the optimization problem of the global score function in Equation 4. On the other hand, it is too expensive to examine the whole solution space for a collection with reasonable size. Here Simulated Annealing (SA) [Kirkpatrick et al 1983, Cerny 1985] is used to search for the global maximum.

Starting from an original state with *n* singleton clusters, the relation matrix randomly changes one element in each round and updates other corresponding relations. If the state change improves the global score in Equation 4, it is kept. Otherwise, the matrix is reverted back to the state before the step with a certain probability. In the earlier stage of the annealing process, the high temperature allows the state to jump out of local optima. But later this algorithm gradually degrades to hill climbing. The SA process continues until the temperature is very low, or the state has not been changed for a certain number of steps. Details of the simulated annealing process can be found in Appendix B.

As SA is non-deterministic and does not always return the same result, multiple runs are often preferred.

5.4 Evaluation

We reuse the story threading data collection [Nallapati et al 2004] with additional annotation in our experiments. 6 topics that belong to the subject “Science and Discovery News” are selected from TDT-3. For these topics, the incidents are already annotated, with a link between each incident pair that shows a contextual relation. For this enhanced framework, each incident is further assigned to one of the following classes: *comment, prediction, damage, background, preparation* or *general*. The annotator also needs to attach a label to each link, where the options can be found in Table 5. Table 8 shows some statistics of the collection. The last three are used in the matrix comparison method at the end of this section.

Science/discovery News	
Topics	6
Total size	280
Topic sizes	52, 43, 158, 77, 2, 6
Language	English
Source	Newswire, broadcast
Incidents	30
N_-	32875
N_0	2583
N_+	3602

Table 8: Experiment Corpus

The evaluation measures in story threading (Section 4.3) are kept here. For the clustering step and the link step, we compare the pair-wise relations between the ground truth and the system output and show the precision and recall. However, with the richer relation type, an additional evaluation criterion is added. In the link evaluation that considers the relation type, a system-assigned link is correct only when it appears at the correct location and its type is the same as the ground truth. After we define relation matrix *R* for the system output and *RT* for the annotation, these evaluation measures are:

$$\begin{aligned}
P_{cluster} &= P(RT_{ij} = 0 \mid R_{ij} = 0) \\
R_{cluster} &= P(R_{ij} = 0 \mid RT_{ij} = 0) \\
P_{bin} &= P(RT_{ij} > 0 \wedge R_{ij} \equiv RT_{ij} (\operatorname{mod} 2) \mid R_{ij} > 0) \\
R_{bin} &= P(R_{ij} > 0 \wedge R_{ij} \equiv RT_{ij} (\operatorname{mod} 2) \mid RT_{ij} > 0) \\
P_{link} &= P(R_{ij} = RT_{ij} \mid R_{ij} > 0) \\
R_{link} &= P(R_{ij} = RT_{ij} \mid RT_{ij} > 0)
\end{aligned}$$

Although defined with the relation matrix, the evaluation measures of clustering and binary link performance are identical to those in Section 4.3 (the correspondence can be found in Appendix C). With the precision and corresponding recall, we also calculate the F-value, which is the harmonic mean of them.

$$\begin{aligned}
F_{cluster} &= \frac{2 \times P_{cluster} \times R_{cluster}}{P_{cluster} + R_{cluster}} \\
F_{bin} &= \frac{2 \times P_{bin} \times R_{bin}}{P_{bin} + R_{bin}} \\
F_{link} &= \frac{2 \times P_{link} \times R_{link}}{P_{link} + R_{link}}
\end{aligned}$$

Note that "cluster" measures score the ability to collect stories into incidents; the "bin" measures capture whether links are found at all (binary); and "link" measures reflect whether the type of a link is also correct. These evaluation measures show the performance of a system on the individual steps. However, it is observed from the experiments that there is usually a tradeoff between the accuracy of clusters and links. When the performance of one improves, the other often decreases with it. A balance is required between these two for an overall evaluation.

Following the observation in Section 4.3, these pair-wise comparison methods also have a magnifying effect of noise. Therefore, evaluation results tend to be small, as they often have a linear relation with the square of the real accuracy.

We also measure the global similarity of *R* and *RT* (the relation matrix for the truth annotation), which shows the overall proximity between the system output and the ground truth. A perfect output should get a relation matrix *R* identical to *RT*, and a mismatch between the corresponding elements of *R* and *RT* means an error. There are three types of values in the relation matrix, -1 (not related), 0 (cluster) and positive (link), so it is natural to compare the allocation of them in the two matrices (*R* and *RT*). As the distribution of these elements in a relation matrix is highly skewed (there are usually many more -1's than the other two types), equal contribution from each position will favor the clustering performance. In the design of the evaluation based on matrix comparison, each class is weighted differently to allow approximately equal contribution.

$$M(R,RT) = \frac{\sum_{i \neq j} match(R_{ij}, RT_{ij})}{\sum_{i \neq j} match(RT_{ij}, RT_{ij})}$$

$$match(x,y) = \begin{cases} 0 & \mathrm{sgn}(x) \neq \mathrm{sgn}(y) \\ 2N_0 / N_+ & x = y > 0 \\ N_0 / N_+ & x > 0, y > 0, x \neq y \\ 1 & x = y = 0 \\ N_0 / N_- & x = y = -1 \end{cases}$$

Here *sgn*(·) is the sign function, which returns 1 if the variable is positive, 0 if it is zero and -1 if negative. N_+ is the number of positive elements in *RT*, N_0 is the number of 0's, and N_- is the number of -1's. Note that *M*(·) is not symmetric, since it biases towards *RT* (the ground truth). With this weight adjustment, the similarity ranges from 0 to 1, where a perfect match returns 1. If a system outputs one big cluster that includes everything, *R* would be all 0's. *match*($R_{ij,} RT_{ij}$) returns 1 for the N_0 zero elements in *RT*, and 0 for others. On the other hand, *match*($RT_{ij,} RT_{ij}$) has N_0 1's, N_+ $2N_0/N_+$'s and N_- N_0/N_-'s. The matrix matching score for such a system is 0.25.

5.5 Experiments

In the experiments, all three algorithms – baseline, two-stage and global optimization are implemented. However, the correlation rules in Table 5 are designed based on personal observations of the whole annotated corpus. Therefore, a strict training/test division cannot be implemented as the training process has already been "contaminated." Results reported below are all performance data on the training set, making them suggestive instead of conclusive.

The evaluation results are shown in Table 9. Some items are marked with an asterisk, which means that the corresponding system is significantly better than the baseline in a one-tailed t-test.

Algorithm	Baseline	Two-stage	Global optimization
$P_{cluster}$	0.548	0.535 (-2.2%)	0.588 (+7.3%)
$R_{cluster}$	0.795	0.924 (+16.2%)	0.632 (-20.4%)
$F_{cluster}$	0.611	0.654 (+7.1%)	0.588 (-3.6%)
P_{bin}	0.164	0.126 (-23.3%)	0.372 (+126.5%)
R_{bin}	0.109	0.048 (-56.2%)	0.447* (+310.1%)
F_{bin}	0.121	0.067 (-44.3%)	0.396 (+227.6%)
P_{link}	0	0.066*	0.156*
R_{link}	0	0.030	0.217*
F_{link}	0	0.040	0.177*
M(R,RT)	0.418	0.442 (+5.5%)	0.519* (+24.0%)

Table 9: Evaluation Results of Three Different Systems: Baseline, Two-stage and Global Optimization (Topic-average)

From the results, introduction of the additional features improves the clustering performance of the two-stage algorithm. However, the application of rules seems to decrease the accuracy in binary links. Failure analysis shows that it is mainly caused by the errors in the label classifier, for which training data are seriously biased towards certain categories (in Table 7, 209 stories in the *General* class for a total of 280). In contrast, global optimization usually returns clusters in slightly lower quality, but the link performance is much higher, especially for the assignment of relation types. Overall, global optimization is regarded as more appropriate for the application since links are very important to form the structure of the incident network, but the lower efficiency restricts its application. The main disadvantage of the baseline and the two-stage algorithm, as shown by failure analysis, is that they cannot correct the clustering errors in later steps. That observation also makes clustering the performance bottleneck.

CHAPTER 6

PASSAGE THREADING

From the beginning of TDT, a one-incident-per-story assumption has been consistently applied. Earlier research provides useful insights in automatic news analysis. However, user experience suggests that the assumption is not always true, although it is effective in reducing the complexity of the problem.

Usually a news story is composed of at least two or three paragraphs, each describing some details of a certain happening or related information. When a user finishes a complete story, it is expected that sufficient context has been included in the story and background information is not always essential (but still beneficial) to understand its content. In short, a news story is often a semantically complete unit.

However, it is not the case for passages[1]. A passage is often short, composed of one or more sentences, and it describes a certain occurrence. For most cases, it is impossible to understand a passage completely without the contextual information from the full story. In an application of passage-based news analysis, this phenomenon directly threatens performance, as the accurate identification of context usually requires semantic understanding of adjacent or even remote passages.

This chapter addresses the problem of analyzing news as smaller text snippets, which is the earliest attempt to conduct incident-based analysis at the passage level. As it is a research area that has not been extensively explored, we do not have any existing data collections that are fully annotated with incident information; the evaluation

[1] A passage is a continuous subset of a news story that contains a complete description of certain news information. It usually follows the natural paragraph or sentence boundaries, but it is also possible that a passage spans across multiple paragraphs.

methods used in TDT or other related research are not directly applicable to this novel application; and we do not know what algorithms work well in the new framework. This chapter starts by investigating supporting issues including data corpora, relevance judgments, and evaluation algorithms. Then we describe two system implementations, including a traditional clustering-threading baseline and a three-stage algorithm that simulates the annotation process. Experiments show significant performance improvement over the baseline when an evaluation measure called cluster-link mean is optimized in the training phase. Finally, a calibration study with hired "users" verifies the value of incident threading in a reading comprehension task, where it proves useful even with the current accuracy of 25-30% in a matrix comparison measure.

6.1 Data Annotation

The first obstacle encountered in passage-level news analysis is the availability of appropriate data collections. There are research topics focusing on finer grained text snippets, including passage retrieval, fact finding, novelty detection, and other similar areas. Some corpora are available for each of these applications, but none of them has provided rich enough annotation that can be directly applied to incident description and contextual analysis. The top priority to prepare for an implementation is to build a data collection sufficiently annotated with reliable relevance judgments, so that the output from algorithms can be compared to the ground truth and an evaluation score will be calculated to measure each algorithm's quality.

Before moving forward to the annotation process, a question must be answered first – instead of complete stories, on what semantic units should we annotate?

For continuous text without explicit structural boundaries, there are several existing methods to separate a stream of text into content-coherent units:

- Salton et al [1996] suggest two strategies of text decomposition: text segments and text themes. A text segment is a group of adjacent paragraphs with strong links (high similarity) among them, and a theme is composed of mutually linked paragraphs often far apart in location.
- TextTiling [Hearst 1994] is another segmentation algorithm that merges pseudo-sentences of pre-defined size. It must decide in advance how many segments a document contains, and the boundaries need to be adjusted to match the actual paragraph break.
- Ponte and Croft [1997] have a different scenario in their segmentation task. The source data come from speech recognition and paragraph boundaries are not available. With sentences as the basic units, the term overlapping is too poor to find enough similarity between them. Therefore, Local Context Analysis (LCA) [Xu and Croft 1996] is applied for term expansion.
- Beeferman et al [1997] introduce a statistical framework based on feature induction. Most of the induced features are individual terms, together with features from short and long-range language models. Each position is assigned a probability if a boundary appears at that point, and many segments do not fall on paragraph breaks.

With these available segmentation methods, there are two choices for the basic unit in annotation. One option is to implement one or more of the algorithms above, and annotate on the generated text segments. However, none of them is 100% accurate, and annotation on an erroneous snippet does not have any value. The other is to follow the built-in breaks in text and assume semantic coherence in each of the structural units. For a text collection without clear semantic boundaries, the first option is preferred. On the contrary, a large proportion of the news corpora available to us are well formatted, at least for newswire reports, in which paragraph and sentence margins are available. To avoid unnecessary noise introduced by segmentation errors, the second option is selected.

Sentences are usually the basic semantic units, but in many occasions the meaning of a sentence cannot be fully understood without any context. That is why expansion is often necessary [Murdock 2006]. From previous observations, the content of a paragraph is coherent in most of the news stories, and the additional information in the longer unit helps to understand the content. Therefore, we opt to treat each paragraph as an independent semantic entity and the basic unit in annotation, which also simplifies the evaluation process.

6.1.1 Data Collection

In Chapters 4 and 5, TDT collections were used (mainly TDT-2 and TDT-3), due to the fact that they contain topical relevance judgments for stories. When TDT was officially terminated at the end of 2004, the latest data (TDT-5) were newswire reports dated between April and September of 2003 [Strassel and Glenn 2004]. As news is very time-sensitive, we believe that recent news reports are more appealing to the readers, thus making them more appropriate for annotation and experiments.

Global Autonomous Language Exploitation (GALE) [Olive 2005] is a DARPA-sponsored research program of which the goal is “to develop and apply computer software technologies to absorb, analyze and interpret huge volumes of speech and text in multiple languages."[1] At the time of this thesis, it is an active topic and new data are released annually for experimental purpose. Its data collections include both news (newswire and broadcast) and web blogs, but the major focus is still on news from various sources. We select documents from the English newswire set only. At the same time, certain queries are issued to collect information on specific subjects, which are slightly different but similar to the traditional TDT topics. The collections in GALE are preferred to TDT because of their up-to-date nature and the ongoing research efforts with these corpora.

```
<query template-number="3" query-id="BAE_TR004">
<query-text>PROVIDE INFORMATION ON [Hassan Hashim Al-Dalimi]</query-text>
<query-arg arg-num="1" arg-type="Person">
<arg-value>Hassan Hashim Al-Dalimi</arg-value>
</query-arg>
<context>
</context>
</query>

<query template-number="11" query-id="BAE_TR012">
<query-text>FIND ACQUAINTANCES OF [Waheed Zaman]</query-text>
<query-arg arg-num="1" arg-type="Person">
<arg-value>Waheed Zaman</arg-value>
</query-arg>
<context>
<location>
<loc> Great Britain</loc>
</location>
</context>
</query>

<query template-number="1" query-id="BAE_TR001">
<query-text>LIST FACTS ABOUT [Bird flu outbreaks in China]</query-text>
<query-arg arg-num="1" arg-type="Event">
<arg-value>Bird flu outbreaks in China</arg-value>
</query-arg>
<context>
</context>
</query>
```

Figure 15: Examples of General Queries in GALE (XML format)

[1] http://www.darpa.mil/ipto/programs/gale/gale.asp as of April 28, 2008.

All queries in GALE are formed from templates. Some templates are general and collect all information related to certain topics or certain person/organizations. Other templates focus more on special scenarios and further limit the range with query arguments. Examples of both types are shown in Figure 15 and Figure 16.

```
<query template-number="5" query-id="BAE_TR009">
<query-text>FIND STATEMENTS MADE BY OR ATTRIBUTED TO [Nina Chahine]
on [Lebanese / Israeli conflict]</query-text>
<query-arg arg-num="1" arg-type="Person">
<arg-value>Nina Chahine</arg-value>
</query-arg>
<query-arg arg-num="2" arg-type="Topic">
<arg-value>Lebanese / Israeli conflict</arg-value>
</query-arg>
<context>
</context>
</query>

<query template-number="7" query-id="BAE_TR011">
<query-text>DESCRIBE THE INVOLVEMENT OF [The United States Department of
State] IN [Evacuating US citizens from Lebanon]</query-text>
<query-arg arg-num="1" arg-type="Organization">
<arg-value>The United States Department of State</arg-value>
</query-arg>
<query-arg arg-num="2" arg-type="Topic">
<arg-value>Evacuating US citizens from Lebanon</arg-value>
</query-arg>
<context>
</context>
</query>

<query template-number="8" query-id="LDC_TR052">
<query-text>DESCRIBE THE PROSECUTION OF [U.S. Representative Tom DeLay]
FOR [money laudering]</query-text>
<query-arg arg-num="1" arg-type="Person">
<arg-value>U.S. Representative Tom DeLay</arg-value>
</query-arg>
<query-arg arg-num="2" arg-type="Crime">
<arg-value>money laundering</arg-value>
</query-arg>
<query-date date-type="Activity">
<start-date>2005-09-01</start-date>
<end-date>2006-08-31</end-date>
</query-date>
<context>
<location>
<loc>Washington, D.C.</loc>
<loc>Texas</loc>
</location>
</context>
</query>
```

Figure 16: Examples of Specific Queries in GALE (XML format)

6.1.2 Annotation Process

The method in which annotation data is collected is that the query is matched to the documents in all appropriate corpora, and the top *n* documents in the ranked list are selected and submitted to one or more annotators. Then the annotators need to go through all paragraphs and filter out those that do not contain description of any real-world happening.

When general queries are submitted to the collection, the top documents returned usually contain news reports in various subjects. Manual analysis shows that many of them are isolated incidents, and do not have any relation with other reports. An incident network generated on such data would include several strongly-connected parts (which correspond to the popular subjects) and a large number of orphan nodes, which is not an effective representation of interesting information. As public interest usually focuses on a few special subjects or topics, we believe that shrinking the scope of news into a single type of information will help improve the coherence between different reports, thus making the contextual analysis more meaningful and easier. After accumulating enough experience within a specific area, it is natural to extend the framework to include other subjects.

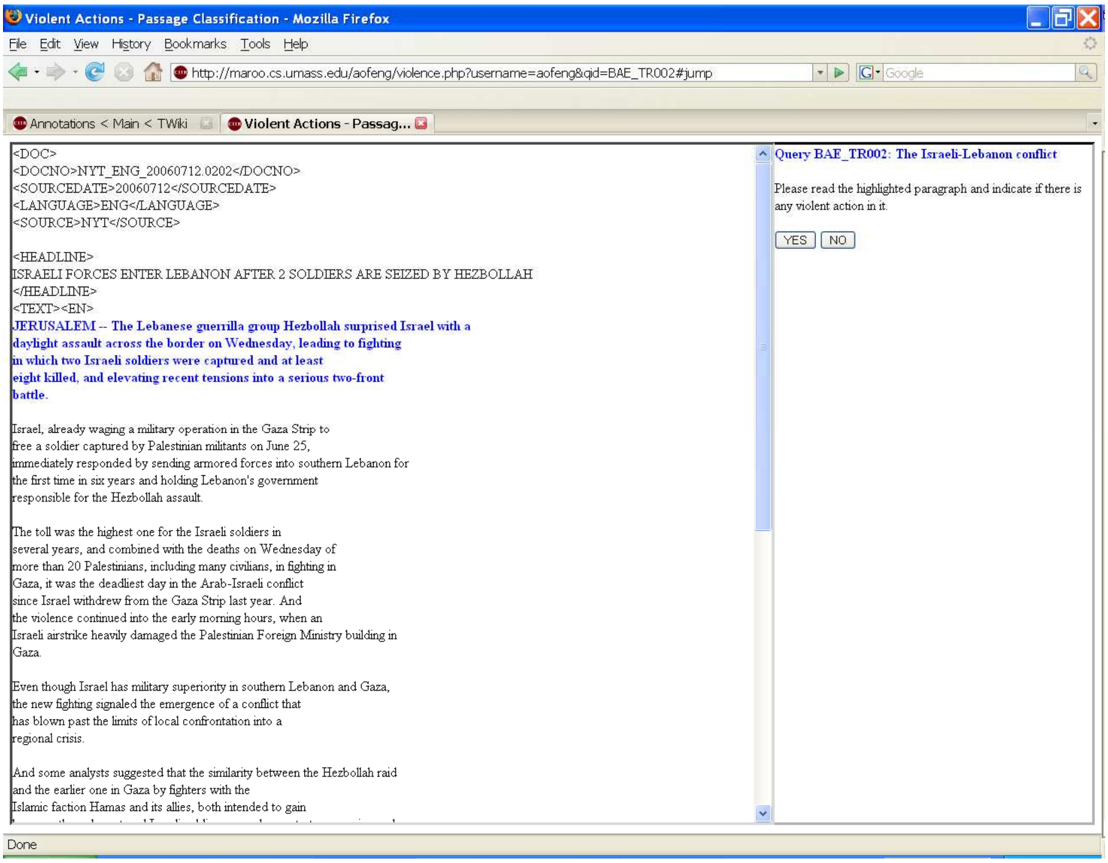
The rest of the chapter will mainly focus on the data annotation and experiments for the specific queries. The general case is discussed in Appendix A.

Within the existing templates of GALE queries, several focus on specific information, and each of them is a good candidate for the first subject of the annotation process. With careful consideration, we select the “strong” activities that involve the description of violent actions. The decision is partially influenced by the public concern about terrorist activities, and it also reflects the general interest in conflicts around the world. The GALE data corpora are rich in both types of information, whereby the selection of this subject is justified. Another reason is that preliminary annotation on this topic shows good inter-annotator consistency.

In the 17 templates available in the second year of GALE program, template 16 - DESCRIBE ATTACKS in [location] GIVING LOCATION (AS SPECIFIC AS POSSIBLE), DATE, AND NUMBER OF DEAD AND INJURED – satisfies the requirement of the violent subject. Other than template 16, there are a small number of queries in other templates that also fall into this category. For example, query LDC_TR008 (LIST FACTS ABOUT [Civil unrest in France]) includes descriptions of many violent activities.

After the selection of a GALE query, it is converted into the Indri [Metzler and Croft 2004] format and matched against the index of English newswire collections. The Indri system returns a ranked list of best matching documents, and the top 10 are selected for annotation purposes.

The annotation process consists of three steps. The first step is for the annotator to walk through each paragraph and identify if it contains any description of a violent action. The second step is to mark the individual violent actions and find co-references of the same activity. The last step scans an incomplete list of incident pairs and the annotator is required to determine if there is any logical or progressional relation in each pair, and mark the direction if such a link exists.


Violent Actions - Passage Classification - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://maroo.cs.umass.edu/aofeng/violence.php?username=aofeng&qid=BAE_TR002#jump
Google
Annotations < Main < TWiki
Violent Actions - Passag...
YES
NO
Done

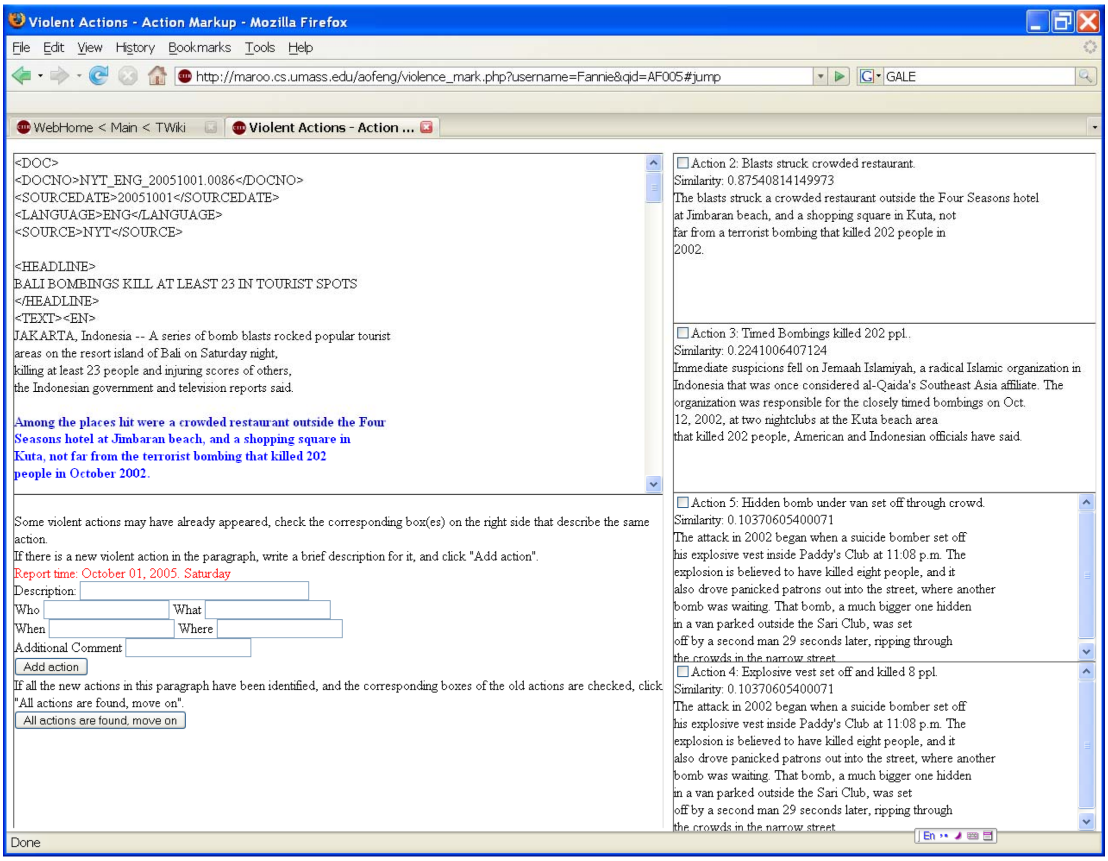
```
<DOC>
<DOCNO>NYT_ENG_20060712.0202</DOCNO>
<SOURCEDATE>20060712</SOURCEDATE>
<LANGUAGE>ENG</LANGUAGE>
<SOURCE>NYT</SOURCE>

<HEADLINE>
ISRAELI FORCES ENTER LEBANON AFTER 2 SOLDIERS ARE SEIZED BY HEZBOLLAH
</HEADLINE>
<TEXT><EN>
JERUSALEM -- The Lebanese guerrilla group Hezbollah surprised Israel with a
daylight assault across the border on Wednesday, leading to fighting
in which two Israeli soldiers were captured and at least
eight killed, and elevating recent tensions into a serious two-front
battle.

Israel, already waging a military operation in the Gaza Strip to
free a soldier captured by Palestinian militants on June 25,
immediately responded by sending armored forces into southern Lebanon for
the first time in six years and holding Lebanon's government
responsible for the Hezbollah assault.

The toll was the highest one for the Israeli soldiers in
several years, and combined with the deaths on Wednesday of
more than 20 Palestinians, including many civilians, in fighting in
Gaza, it was the deadliest day in the Arab-Israeli conflict
since Israel withdrew from the Gaza Strip last year. And
the violence continued into the early morning hours, when an
Israeli airstrike heavily damaged the Palestinian Foreign Ministry building in
Gaza.

Even though Israel has military superiority in southern Lebanon and Gaza,
the new fighting signaled the emergence of a conflict that
has blown past the limits of local confrontation into a
regional crisis.

And some analysts suggested that the similarity between the Hezbollah raid
and the earlier one in Gaza by fighters with the
Islamic faction Hamas and its allies, both intended to gain
```

Query BAE_TR002: The Israeli-Lebanon conflict

Please read the highlighted paragraph and indicate if there is any violent action in it.

Figure 17: Annotation Interface of Step 1

The first step starts with a login screen, where the annotator needs to input the user name and select a query from a drop-down list. After logging into the annotation system, the user interface looks like Figure 17. On the left side of the interface, the whole document is displayed, with one paragraph highlighted, on which is the current paragraph to work. The annotator should read the paragraph carefully and decide if there is any description of violent actions in it. Here *violence* refers to an act of aggression that causes or intends to cause injury to person(s) or property, including terrorist, military or other types of violent behavior. The *action* means the aggression itself, not including comments, follow-ups (investigation, arrest, trial, revenge, and other related actions, unless they also involve some type of violence), and analyses. A general description about a violence subject does not qualify as an action. There are only two options for each paragraph, either YES (there is at least one description of violent actions in the paragraph) or NO (there does not exist any description of violent actions in the paragraph). It is not uncommon to see paragraphs that contain multiple violent activities.


Violent Actions - Action Markup - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://maroo.cs.umass.edu/aofeng/violence_mark.php?username=Fannie&qid=AF005#jump
GALE
WebHome < Main < TWiki
Violent Actions - Action ...
Done
En

```
<DOC>
<DOCNO>NYT_ENG_20051001.0086</DOCNO>
<SOURCEDATE>20051001</SOURCEDATE>
<LANGUAGE>ENG</LANGUAGE>
<SOURCE>NYT</SOURCE>

<HEADLINE>
BALI BOMBINGS KILL AT LEAST 23 IN TOURIST SPOTS
</HEADLINE>
<TEXT><EN>
JAKARTA, Indonesia -- A series of bomb blasts rocked popular tourist
areas on the resort island of Bali on Saturday night,
killing at least 23 people and injuring scores of others,
the Indonesian government and television reports said.

Among the places hit were a crowded restaurant outside the Four
Seasons hotel at Jimbaran beach, and a shopping square in
Kuta, not far from the terrorist bombing that killed 202
people in October 2002.
```

Some violent actions may have already appeared, check the corresponding box(es) on the right side that describe the same action.

If there is a new violent action in the paragraph, write a brief description for it, and click "Add action".

Report time: October 01, 2005. Saturday

Description:

Who What

When Where

Additional Comment

Add action

If all the new actions in this paragraph have been identified, and the corresponding boxes of the old actions are checked, click "All actions are found, move on".

All actions are found, move on

□ Action 2: Blasts struck crowded restaurant.
Similarity: 0.87540814149973
The blasts struck a crowded restaurant outside the Four Seasons hotel
at Jimbaran beach, and a shopping square in Kuta, not
far from a terrorist bombing that killed 202 people in
2002.

□ Action 3: Timed Bombings killed 202 ppl..
Similarity: 0.2241006407124
Immediate suspicions fell on Jemaah Islamiyah, a radical Islamic organization in
Indonesia that was once considered al-Qaida's Southeast Asia affiliate. The
organization was responsible for the closely timed bombings on Oct.
12, 2002, at two nightclubs at the Kuta beach area
that killed 202 people, American and Indonesian officials have said.

□ Action 5: Hidden bomb under van set off through crowd.
Similarity: 0.10370605400071
The attack in 2002 began when a suicide bomber set off
his explosive vest inside Paddy's Club at 11:08 p.m. The
explosion is believed to have killed eight people, and it
also drove panicked patrons out into the street, where another
bomb was waiting. That bomb, a much bigger one hidden
in a van parked outside the Sari Club, was set
off by a second man 29 seconds later, ripping through
the crowds in the narrow street

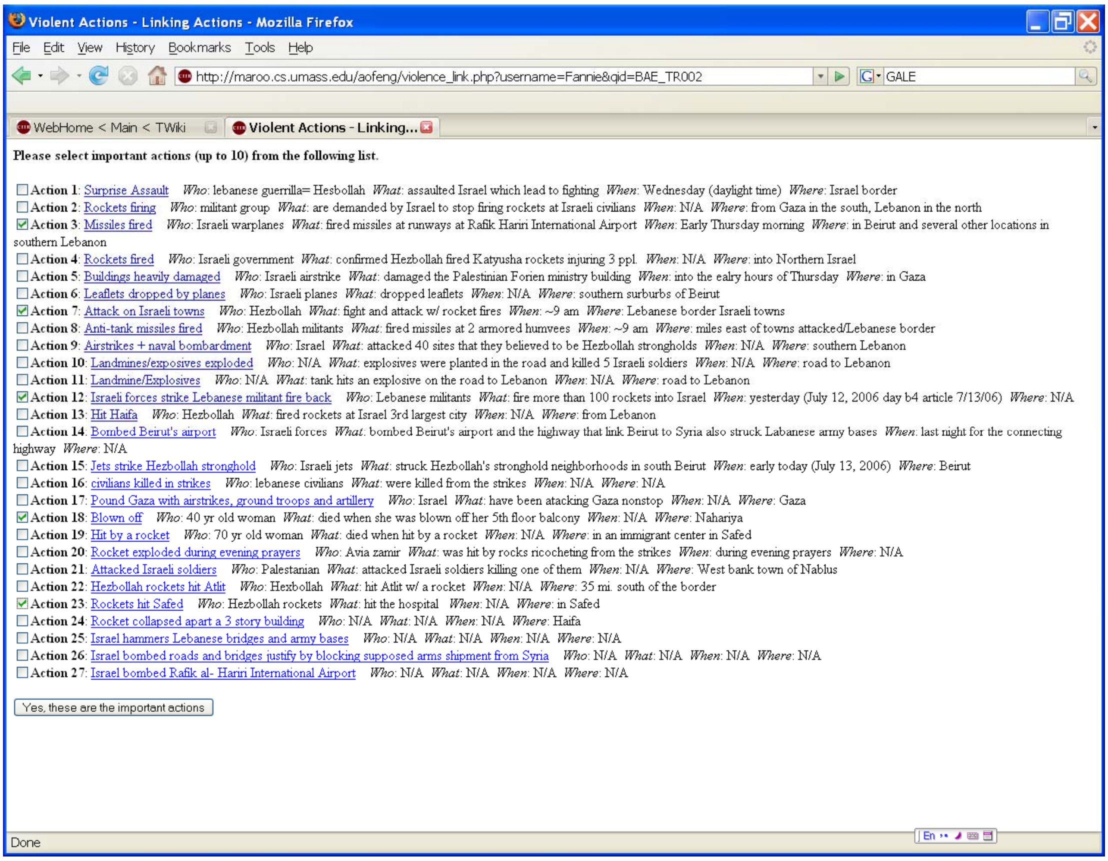
□ Action 4: Explosive vest set off and killed 8 ppl.
Similarity: 0.10370605400071
The attack in 2002 began when a suicide bomber set off
his explosive vest inside Paddy's Club at 11:08 p.m. The
explosion is believed to have killed eight people, and it
also drove panicked patrons out into the street, where another
bomb was waiting. That bomb, a much bigger one hidden
in a van parked outside the Sari Club, was set
off by a second man 29 seconds later, ripping through
the crowds in the narrow street

Figure 18: Annotation Interface of Step 2

After all the paragraphs in the top 10 documents have been finished in Step 1, the annotation automatically continues to the second step. As shown in Figure 18, the document is again displayed on the left side, but in a smaller window. The current paragraph is also highlighted, but there are more operations in this step. Only paragraphs identified as “containing at least one violent action” are highlighted in this step, and others are automatically skipped as they do not contain any interesting information. For the violent actions available in the highlighted paragraph, there are two possibilities:

1. The violent activity that it describes has not been observed in the previous reports. Then it is identified as a new incident. For this case, a description is required for the incident, and other attributes (who, when, where, what) should also be provided when available. After filling in the form in the bottom left corner, the "Add action" button should be clicked and the new incident will appear in the list under that button, with a checked box in front of it, which indicates that this incident is described in the current paragraph.
2. The violent activity has been described before, and the previous description should have been identified in an incident. On the right side of the annotation interface, up to four earlier incidents are listed together with the corresponding text, and they are sorted by their similarity to the current paragraph. Under most conditions, the previous incident about which the paragraph talks appears in the list, and the annotator will check the box in front of it, which identifies the content of the current paragraph.

For cases where multiple violent actions appear in the same paragraph, it is likely to see both new and old incidents. Under that condition, both instructions above apply to the paragraph. After all new incidents have been annotated and all old incidents marked, the annotator clicks on "All actions are found, move on" button. The judgment of the current paragraph is then stored and the focus moves to the next one.

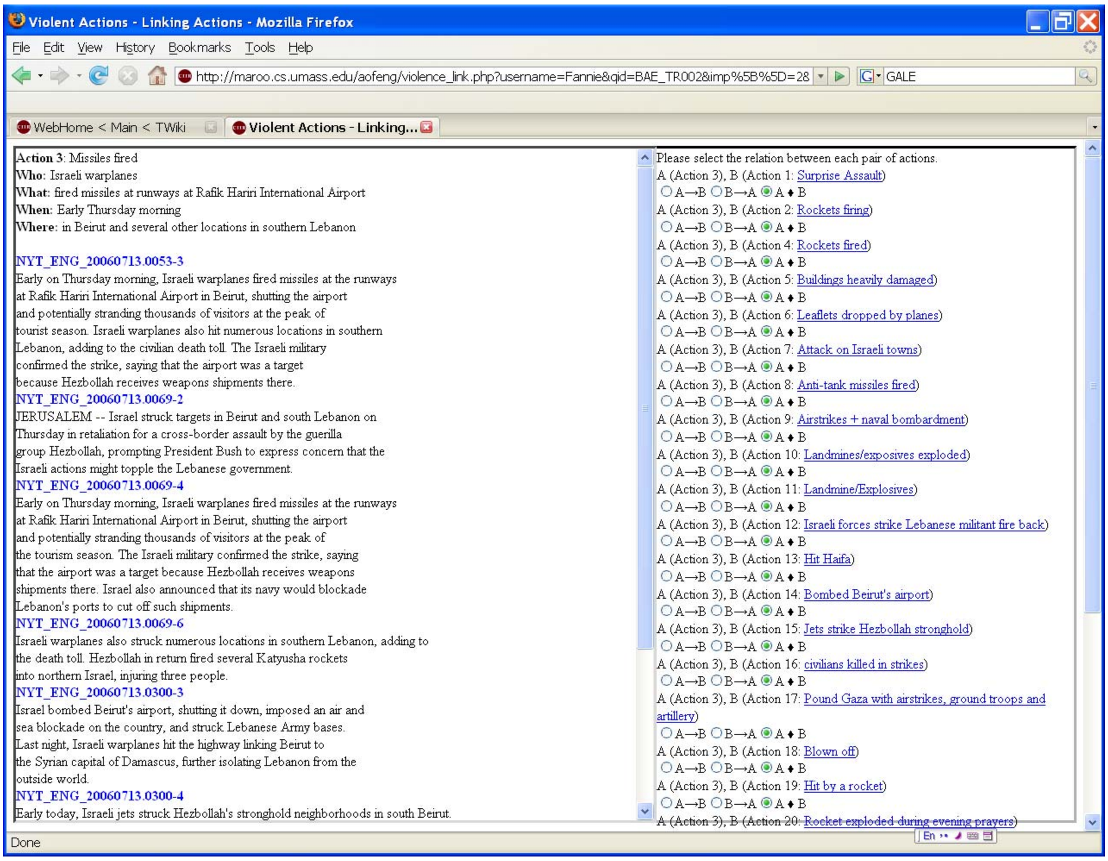

Violent Actions - Linking Actions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://maroo.cs.umass.edu/aofeng/violence_link.php?username=Fannie&qid=BAE_TR002
GALE
WebHome < Main < TWiki
Violent Actions - Linking...
Done

Please select important actions (up to 10) from the following list.

☐ **Action 1**: Surprise Assault *Who*: lebanese guerrilla= Hesbollah *What*: assaulted Israel which lead to fighting *When*: Wednesday (daylight time) *Where*: Israel border
☐ **Action 2**: Rockets firing *Who*: militant group *What*: are demanded by Israel to stop firing rockets at Israeli civilians *When*: N/A *Where*: from Gaza in the south, Lebanon in the north
☑ **Action 3**: Missiles fired *Who*: Israeli warplanes *What*: fired missiles at runways at Rafik Hariri International Airport *When*: Early Thursday morning *Where*: in Beirut and several other locations in southern Lebanon
☐ **Action 4**: Rockets fired *Who*: Israeli government *What*: confirmed Hezbollah fired Katyusha rockets injuring 3 ppl. *When*: N/A *Where*: into Northern Israel
☐ **Action 5**: Buildings heavily damaged *Who*: Israeli airstrike *What*: damaged the Palestinian Forien ministry building *When*: into the ealry hours of Thursday *Where*: in Gaza
☐ **Action 6**: Leaflets dropped by planes *Who*: Israeli planes *What*: dropped leaflets *When*: N/A *Where*: southern surburbs of Beirut
☑ **Action 7**: Attack on Israeli towns *Who*: Hezbollah *What*: fight and attack w/ rocket fires *When*: ~9 am *Where*: Lebanese border Israeli towns
☐ **Action 8**: Anti-tank missiles fired *Who*: Hezbollah militants *What*: fired missiles at 2 armored humvees *When*: ~9 am *Where*: miles east of towns attacked/Lebanese border
☐ **Action 9**: Airstrikes + naval bombardment *Who*: Israel *What*: attacked 40 sites that they believed to be Hezbollah strongholds *When*: N/A *Where*: southern Lebanon
☐ **Action 10**: Landmines/exposives exploded *Who*: N/A *What*: explosives were planted in the road and killed 5 Israeli soldiers *When*: N/A *Where*: road to Lebanon
☐ **Action 11**: Landmine/Explosives *Who*: N/A *What*: tank hits an explosive on the road to Lebanon *When*: N/A *Where*: road to Lebanon
☑ **Action 12**: Israeli forces strike Lebanese militant fire back *Who*: Lebanese militants *What*: fire more than 100 rockets into Israel *When*: yesterday (July 12, 2006 day b4 article 7/13/06) *Where*: N/A
☐ **Action 13**: Hit Haifa *Who*: Hezbollah *What*: fired rockets at Israel 3rd largest city *When*: N/A *Where*: from Lebanon
☐ **Action 14**: Bombed Beirut's airport *Who*: Israeli forces *What*: bombed Beirut's airport and the highway that link Beirut to Syria also struck Labanese army bases *When*: last night for the connecting highway *Where*: N/A
☐ **Action 15**: Jets strike Hezbollah stronghold *Who*: Israeli jets *What*: struck Hezbollah's stronghold neighborhoods in south Beirut *When*: early today (July 13, 2006) *Where*: Beirut
☐ **Action 16**: civilians killed in strikes *Who*: lebanese civilians *What*: were killed from the strikes *When*: N/A *Where*: N/A
☐ **Action 17**: Pound Gaza with airstrikes, ground troops and artillery *Who*: Israel *What*: have been atacking Gaza nonstop *When*: N/A *Where*: Gaza
☑ **Action 18**: Blown off *Who*: 40 yr old woman *What*: died when she was blown off her 5th floor balcony *When*: N/A *Where*: Nahariya
☐ **Action 19**: Hit by a rocket *Who*: 70 yr old woman *What*: died when hit by a rocket *When*: N/A *Where*: in an immigrant center in Safed
☐ **Action 20**: Rocket exploded during evening prayers *Who*: Avia zamir *What*: was hit by rocks ricocheting from the strikes *When*: during evening prayers *Where*: N/A
☐ **Action 21**: Attacked Israeli soldiers *Who*: Palestanian *What*: attacked Israeli soldiers killing one of them *When*: N/A *Where*: West bank town of Nablus
☐ **Action 22**: Hezbollah rockets hit Atlit *Who*: Hexbollah *What*: hit Atlit w/ a rocket *When*: N/A *Where*: 35 mi. south of the border
☑ **Action 23**: Rockets hit Safed *Who*: Hezbollah rockets *What*: hit the hospital *When*: N/A *Where*: in Safed
☐ **Action 24**: Rocket collapsed apart a 3 story building *Who*: N/A *What*: N/A *When*: N/A *Where*: Haifa
☐ **Action 25**: Israel hammers Lebanese bridges and army bases *Who*: N/A *What*: N/A *When*: N/A *Where*: N/A
☐ **Action 26**: Israel bombed roads and bridges justify by blocking supposed arms shipment from Syria *Who*: N/A *What*: N/A *When*: N/A *Where*: N/A
☐ **Action 27**: Israel bombed Rafik al- Hariri International Airport *Who*: N/A *What*: N/A *When*: N/A *Where*: N/A

Figure 19: Annotation Interface of Step 3 – Important Incident Selection

Again, the annotation interface directly jumps to the third step when all paragraphs are finished in step 2. The first screen after logging in is shown in Figure 19. As there are often a large number of incidents for each query, it is an exhaustive task to annotate the relations for all possible incident pairs. Here a smaller number of important incidents (those core activities that represent the main occurrences in the whole topic) are selected, and only the relations between these incidents and others need to be marked, which is usually much cheaper than comparing all pairs. As we can see, each incident comes with its description and all the features, if they have been identified in the previous step. In order to remind the annotator of the content for each incident, the description is also a super link which opens a window that displays all the corresponding text for that incident.


Violent Actions - Linking Actions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://maroo.cs.umass.edu/aofeng/violence_link.php?username=Fannie&qid=BAE_TR002&imp%5B%5D=28
GALE
WebHome < Main < TWiki
Violent Actions - Linking...
Done
En

Action 3: Missiles fired
Who: Israeli warplanes
What: fired missiles at runways at Rafik Hariri International Airport
When: Early Thursday morning
Where: in Beirut and several other locations in southern Lebanon

NYT_ENG_20060713.0053-3
Early on Thursday morning, Israeli warplanes fired missiles at the runways at Rafik Hariri International Airport in Beirut, shutting the airport and potentially stranding thousands of visitors at the peak of tourist season. Israeli warplanes also hit numerous locations in southern Lebanon, adding to the civilian death toll. The Israeli military confirmed the strike, saying that the airport was a target because Hezbollah receives weapons shipments there.
NYT_ENG_20060713.0069-2
JERUSALEM -- Israel struck targets in Beirut and south Lebanon on Thursday in retaliation for a cross-border assault by the guerilla group Hezbollah, prompting President Bush to express concern that the Israeli actions might topple the Lebanese government.
NYT_ENG_20060713.0069-4
Early on Thursday morning, Israeli warplanes fired missiles at the runways at Rafik Hariri International Airport in Beirut, shutting the airport and potentially stranding thousands of visitors at the peak of the tourism season. The Israeli military confirmed the strike, saying that the airport was a target because Hezbollah receives weapons shipments there. Israel also announced that its navy would blockade Lebanon's ports to cut off such shipments.
NYT_ENG_20060713.0069-6
Israeli warplanes also struck numerous locations in southern Lebanon, adding to the death toll. Hezbollah in return fired several Katyusha rockets into northern Israel, injuring three people.
NYT_ENG_20060713.0300-3
Israel bombed Beirut's airport, shutting it down, imposed an air and sea blockade on the country, and struck Lebanese Army bases. Last night, Israeli warplanes hit the highway linking Beirut to the Syrian capital of Damascus, further isolating Lebanon from the outside world.
NYT_ENG_20060713.0300-4
Early today, Israeli jets struck Hezbollah's stronghold neighborhoods in south Beirut.

Please select the relation between each pair of actions.
A (Action 3), B (Action 1: Surprise Assault)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 2: Rockets firing)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 4: Rockets fired)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 5: Buildings heavily damaged)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 6: Leaflets dropped by planes)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 7: Attack on Israeli towns)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 8: Anti-tank missiles fired)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 9: Airstrikes + naval bombardment)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 10: Landmines/exposives exploded)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 11: Landmine/Explosives)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 12: Israeli forces strike Lebanese militant fire back)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 13: Hit Haifa)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 14: Bombed Beirut's airport)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 15: Jets strike Hezbollah stronghold)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 16: civilians killed in strikes)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 17: Pound Gaza with airstrikes, ground troops and artillery)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 18: Blown off)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 19: Hit by a rocket)
○ A→B ○ B→A ● A ♦ B
A (Action 3), B (Action 20: Rocket exploded during evening prayers)

Figure 20: Annotation Interface of Step 3 – Relation Markup

With the selection of key incidents, the annotator would see the next phrase of the third step – relation markup. The interface is presented in Figure 20. The left side shows one of the selected important activities, its attributes, and all text snippets associated with it. On the right side, there is an item for each of the other incidents, and the relation between them needs to be annotated. There are three options for the annotator:

1. They are not related, which is the default value.
2. They are connected by a logical or progressional relation, and the incident on the left side is the earlier one.

3. They are connected by a logical or progressional relation, and the incident on the left side is the later one.

One may wonder why the relation types discussed in Chapter 5 are not part of the annotation task. Since only the descriptions of violent actions are considered relevant and related issues (public comments, analysis, government reactions, and preventive actions) are ignored, many of the logical relation types will not appear in this framework. Another reason is that annotators often have difficulty in telling the "correct" relation type, while each person has a different standard for those relations. Therefore, the current design settles on a binary link, which implies the consequence or progressional relation for most cases. If a broader range of incidents is included, distinguishing between various relation types will be desirable.

Due to the lack of rich relations, the global optimization framework is also inappropriate for the current setting of passage threading. When measuring how well a passage pair fits the rule of a specific relation, the relation type is the most important factor in the design of the formula. Therefore, omission of the type increases the difficulty of forming an appropriate link function, as a generic relation includes many possibilities, each involving a different requirement for the two passages. Without an accurate formula to model the dependency, global optimization may not work.

From the statistics, an average of approximately 3 hours is spent on all three steps of a single query, but it highly depends on the speed of the individual annotator and the amount of violence information in each query. Currently 17 queries have been completely annotated through this process. Within these queries, 11 are taken directly from the second year evaluation of GALE, while the other 6 are designed in a similar way to the GALE queries, and the arguments in them are extracted from parts of the corpora that the GALE queries have missed.

For a story describing the Israeli-Lebanon conflict (Figure 21), the annotation outputs of the three steps are shown in Figures 22, 23 and 24, respectively. Since the previous incident analysis of this story is not limited to violent information, all incidents are annotated in this example.

(a) JERUSALEM -- The Lebanese guerrilla group Hezbollah surprised Israel with a daylight assault across the border on Wednesday, leading to fighting in which two Israeli soldiers were captured and at least eight killed, and elevating recent tensions into a serious two-front battle.
(b) Israel, already waging a military operation in the Gaza Strip to free a soldier captured by Palestinian militants on June 25, immediately responded by sending armored forces into southern Lebanon for the first time in six years and holding Lebanon's government responsible for the Hezbollah assault.
(c) The toll was the highest one for the Israeli soldiers in several years, and combined with the deaths on Wednesday of more than 20 Palestinians, including many civilians, in fighting in Gaza, it was the deadliest day in the Arab-Israeli conflict since Israel withdrew from the Gaza Strip last year. Andthe violence continued into the early morning hours, when an Israeli airstrike heavily damaged the Palestinian Foreign Ministry building in Gaza.
(d) Even though Israel has military superiority in southern Lebanon and Gaza, the new fighting signaled the emergence of a conflict that has blown past the limits of local confrontation into a regional crisis.
(e) And some analysts suggested that the similarity between the Hezbollah raid and the earlier one in Gaza by fighters with the Islamic faction Hamas and its allies, both intended to gain leverage through captured Israeli soldiers, may demonstrate a growing and troubling rapport between the two groups.
(f) As with the Gaza conflict, Israel ruled out negotiations with the Lebanese captors of the Israeli soldiers. Prime Minister Ehud Olmert said he held the Lebanese government responsible for the assault by Hezbollah, the Shiite Muslim group that participates in Lebanese politics but also continues to battle Israel.
(g) "I want to make clear that the event this morning is not a terror act, but an act of a sovereign state that attacked Israel without reason," Olmert said. "The government of Lebanon, of which Hezbollah is a part, is trying to shake the stability of the region."
(h) Israel is demanding that its soldiers be returned unconditionally and that militant groups stop firing rockets at Israeli civilians from Gaza in the south and Lebanon in the north.
(i) But both Hamas and Hezbollah are holding out for an exchange for a large number of Palestinian and other Arab prisoners held by Israel.
(j) "The prisoners will not be returned except through one way -- indirect negotiations and a trade," said the leader of Hezbollah, Sheik Hassan Nasrallah, speaking to reporters in Beirut late on Wednesday.

Figure 21: Sample News Story - NYT_ENG_20060712.0202 (copy of Figure 2)

The output of the first step (Figure 22) is composed of two columns. The first column is the paragraph ID, and the second one marks if there is an incident in this paragraph (two options: YES or NO).

```
NYT_ENG_20060712.0202.a          YES
NYT_ENG_20060712.0202.b          YES
NYT_ENG_20060712.0202.c          YES
NYT_ENG_20060712.0202.d          NO
NYT_ENG_20060712.0202.e          NO
NYT_ENG_20060712.0202.f          YES
NYT_ENG_20060712.0202.g          YES
NYT_ENG_20060712.0202.h          YES
NYT_ENG_20060712.0202.i          YES
NYT_ENG_20060712.0202.j          YES
```

Figure 22: Sample Output of Step 1

```
NYT_ENG_20060712.0202.a          0
NYT_ENG_20060712.0202.b          1
NYT_ENG_20060712.0202.c          2
NYT_ENG_20060712.0202.d          NO
NYT_ENG_20060712.0202.e          NO
NYT_ENG_20060712.0202.f          3
NYT_ENG_20060712.0202.g          3
NYT_ENG_20060712.0202.h          3
NYT_ENG_20060712.0202.i          4
NYT_ENG_20060712.0202.j          4

0     Wednesday      border            Hezbollah            assault
Hezbollah conducts surprise attack towards Israel.

1     immediately    Southern Lebanon  Israel               send
Israel sends troops to southern Lebanon.

2     early morning  Gaza              Israel               air strike
Israel bombs the Palestinian Foreign Ministry building.

3     N/A            N/A               Israel               rule out
Israel refuses negotiations with Hezbollah.

4     N/A            N/A               Hamas and Hezbollah  hold out
Hamas and Hezbollah request for a prisoner exchange.
```

Figure 23: Sample Output of Step 2

Step 2 outputs similar data to the previous step, but a paragraph is marked with the incident number instead. In addition to the paragraph annotation, there is a separate file that contains the description and main features of each incident (Figure 23).

The result of Step 3 (Figure 24) has two columns, where each row represents a link. The first element in a row is the source incident of the link (the premise in a logical relation or the earlier incident in a progressional relation), and the second is its destination. The important incidents in this story are 0 (Hezbollah conducts surprise attack towards Israel) and 1 (Israel sends troops to southern Lebanon).

```
0        1
0        3
0        4
```

Figure 24: Sample Output of Step 3

Although the story in Figure 21 is a toy example that annotates other non-violent incidents, the real annotation output of a violent query is very similar to it, except that non-violent paragraphs are always marked *NO*.

Statistics of the annotated corpus are displayed in Table 10. For each line (except the number of queries), we show the total number of objects as well as the minimum and maximum in 17 queries.

Queries	17
Documents	170 (10 – 10)
Passages	3,618 (101 – 277)
“violent” passages	792 (10 – 93)
Percentage of “violent” passages	21.8% (6.6 – 70.2%)
Incidents	376 (4 – 45)
Links	156 (0 – 47)

Table 10: Statistics of Annotated Corpus in Passage Threading

6.1.3 Inter-annotator Agreement

While it is important to hire annotators and have them finish the relevance judgment for certain collections, any task dealing with natural languages is subjective and unexpected errors or disagreements are often associates with it. The quality of the annotation is usually difficult to evaluate directly, as universal truth does not exist for such an application. What people usually do is to have multiple annotators work on the same dataset, and then calculate their inter-annotator agreement afterwards. If the agreement is high, the task is considered well defined. Otherwise, it is generally necessary to revise the framework or instruction to improve clarity.

In the three annotation steps described above, the second step involves subjective description of violent actions, and previous experience shows that variance in granularity is very common among different people. The last step seems to be more objective as the judgment is a simple selection among three options. Unfortunately, the links are derived from the incidents created in the previous step. While some overlap of annotated incidents is observed, it is difficulty to reach an agreement on the granularity issue. Therefore, it is rare to see an identical link assignment between two equivalent incidents from different annotators, making it almost impossible to directly compare their judgments in this step. Only the first step is appropriate for comparison, as everyone is working on the same objects (paragraphs in the top 10 documents), and choices are very limited for each (yes or no).

For an annotation with more than two raters, Fleiss' Kappa coefficient [Fleiss 1971] is commonly used. The ideal agreement among a certain number of annotators is 1, in which everyone makes exactly the same judgment for all test cases. If each annotator only pretends to read the material, but in fact randomly selects an answer based on known prior probabilities of those options, the agreement would be P_e. Given that the actual agreement is P_a, the Fleiss' Kappa is,

$$\kappa_F = \frac{P_a - P_e}{1 - P_e}$$

Cohen's Kappa coefficient [Cohen 1960] has a similar equation to Fleiss' Kappa, except that it only deals with the case when two annotators are compared. Instead of the distribution-based agreement calculation in Fleiss', Cohen's Kappa takes the simple percentage of agreement among all test cases.

As passage-level annotation is a time-consuming task, we did not require every annotator to finish all queries. We selected 5 queries and assigned them to 5 annotators, and each of them was asked to finish the first step of all queries. As some annotators did not complete the whole task, the final inter-annotator agreement was calculated based on 4 queries and 4 annotators.

For Fleiss' Kappa,

$$P_a = 0.813$$
$$P_e = 0.538$$
$$\kappa_F = 0.595$$

All the pair-wise comparisons (Cohen's Kappa) of these annotators are listed in Table 11.

Cohen's Kappa	Annotator 1	Annotator 2	Annotator 3	Annotator 4	Average
Annotator 1	-	0.572	0.551	0.664	0.596
Annotator 2	0.572	-	0.548	0.630	0.583
Annotator 3	0.551	0.548	-	0.618	0.572
Annotator 4	0.664	0.630	0.618	-	0.637

Table 11: Inter-annotator Agreement for Violent Actions

There is no general rule what Kappa value means a good agreement. Depending on the number of categories and subjects, the same Kappa value may correspond to different levels of inter-annotator agreement. Landis and Koch [1977] provide the following table for reference, but no evidence is supplied to support the claim.

κ	Interpretation
< 0	No agreement
0.0 – 0.20	Very low agreement
0.21 – 0.40	Low agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Full agreement
0.81 – 1.00	Almost perfect agreement

Table 12: Approximate Explanation for Kappa Values

Despite the obscurity of Kappa values, the result of the calculation shows good agreement among annotators, and it would be safe to claim that the problem definition is clear enough for the annotators to make sensible choices.

In addition to the overall agreement among annotators, it is also interesting how well they agree with each other on a certain class of answers. In the first annotation step, there are only two choices, yes or no. So the probability of agreement can be easily calculated. When one annotator says "yes" or "no" for a specific paragraph, the chance that another annotator will make the same judgment is a probability describing their agreement. The result from the same dataset is:

P (B says yes | A says yes) = 0.741

P (B says no | A says no) = 0.854

The chance of agreeing on a "no" answer is higher. It is not a surprise because there are more paragraphs in the collection that do not contain any violent action than those that do.

6.2 Evaluation

As observed from the previous section, the top-ranked documents returned by a query are composed of multiple incidents, each containing one or more text snippets, and links go between certain incident pairs to show the existence of their relation. The system output is similar to the ground truth, while the passages form clusters with links connecting them. However, how to compare the system output to the annotation is a nontrivial problem, as a strict map between the truth incidents and the system clusters does not exist under most conditions.

6.2.1 Attributes of a Preferred System

If there are several algorithms that all run on the same data collection and output different results, we need a comparison with the ground truth, if it exists, to decide their quality. Since there are not any established evaluation measures for this application, we can start from the attributes that a user would like to see in a "good" system.

1. People have difficulty understanding too many things at the same time. If a system outputs hundreds of clusters and thousands of links, it is almost impossible to comprehend the useful information in this complex network. Therefore, cleaner outputs are preferred.
2. When a user looks at a cluster in the system output, he/she would expect that most of the text snippets in it have the same (or at least similar) content. If the cluster is not "pure," it has low quality.
3. If the ground truth is available for an incident, all the snippets in it are known to be about the same occurrence. In the system output, we expect that these passages

would appear in the same cluster. If the ideal case cannot be achieved, at least it is not desirable that they are distributed all over the incident network. The degree of concentration is a good indication for the quality of an incident.

4. Talking about links, they usually exist between two incidents that do not look very similar but contain some logical or temporal relation. Linking two passages that are about the same incident is a mistake, and connecting two unrelated actions is also an error. It is desirable that the system-generated links go between the passage pairs where and only where the ground truth indicates a logical or temporal relation.
5. A link is directional, as it directly models the causal or progressional relation between the corresponding incidents. If a link is found at the correct location but it comes in the wrong direction, it is still a mistake. Usually such an error is punished less severely than the case that it is not generated while the ground truth indicates otherwise.

With these attributes, some evaluation measures can be naturally defined, with each reflecting the quality of a certain aspect.

6.2.2 Evaluation Criteria

The evaluation is mainly composed of two parts, each corresponding to one portion of the implementation process. The first part evaluates the clustering step, which measures how similar the incidents and the system clusters are. The second focuses on links, mainly on the overlap between the links that appear in the annotation and those in the system output. However, evaluations in these two are partially independent and cannot provide a single score for system comparison. For a good estimate of the overall performance, these individual measures are combined to generate a score.

6.2.2.1 Clustering Evaluation

Suppose that an incident *I* includes *p* passages from the annotation. In the system output, there are *n* clusters, and the numbers of passages in each cluster which belong to incident *I* are $p_1, p_2, \ldots, p_n$, respectively. These numbers add up to *p*. The concentration for incident *I* is defined as:

$$Conc(I) = \frac{\sum_{i=1}^{n} p_i(p_i - 1)}{p(p-1)}$$

From the equation, concentration measures how "concentrated" the passages in an incident are. When all *p* passages appear in the same cluster, the concentration is 1. If they are evenly distributed into two clusters, the score is approximately 0.5 when p is large enough. If the *p* passages are distributed in *p* different clusters, the calculation returns 0. The equation is borrowed from the calculation in Fleiss' Kappa [Fleiss 1971], and it only works when $p > 1$ (otherwise the denominator is 0). The concentration score can be calculated for all incidents with size larger than 1, and an average of them is taken based on the size:

$$Concentration = \frac{\sum_i conc(I_i)|I_i|}{\sum_i |I_i|}$$

Concentration itself is not good enough for evaluation purposes. When all passages are assigned to the same cluster, the concentration score is always 1, but this huge cluster will be heterogeneous. Thus another measure is necessary for the quality of clusters.

In a cluster *C* with size *q*, its members may be snippets in different incidents from the annotation. Suppose that *m* incidents exist, and the numbers of *C*'s members belonging to each of them are $q_1, q_2, \ldots, q_m$, respectively. These numbers may not add up to *q*, because there are passages that do not belong to any incident[1]. Similarly to the incident concentration, the cluster purity of *C* can be defined as:

$$Pur(C) = \frac{\sum_{i=1}^{m} q_i(q_i - 1)}{q(q-1)}$$

Similarly to concentration, the purity score measures how "pure" each cluster is. If everything in a cluster belongs to the same incident, the score is 1. On the contrary, if we cannot find two passages in a cluster from the same incident, the purity is 0. Of course, *q* cannot be 1, otherwise the denominator becomes 0. Averaging overall non-singleton clusters generates the purity score:

$$Purity = \frac{\sum_{i} Pur(C_i)|C_i|}{\sum_{i} |C_i|}$$

Concentration and purity are both scores to evaluate the quality of the system clusters. For the same system, the parameter setting changes the performance, but these two measures are usually negatively-correlated, i.e., the increase of one often leads to the decrease of the other.

[1] The paragraphs with no description of violent actions are marked "NO" in Step 1 of the annotation process, and they will not be assigned to any incident in Step 2.

Although incident concentration and cluster purity are calculated in a different way in comparison to the traditional evaluation measures, they are, to some extent, related to the pair-wise clustering precision and recall [Nallapati et al 2004]. If an incident has high concentration, the majority of passages in it should belong to one or two clusters, therefore most passage pairs in the incident can also be observed in the same clusters in the system output. It means high pair-wise clustering recall. Similarity, cluster purity is positively-correlated with pair-wise clustering precision. For comparison with previous experiments, we keep these evaluation measures (clustering precision – CP, clustering recall – CR) in the clustering phase.

CP = P(p_i, p_j in the same truth incident | p_i, p_j in the same system cluster)

CR = P(p_i, p_j in the same system cluster | p_i, p_j in the same truth incident)

The only change is that p_i and p_j are passages instead of stories, as they were in story threading.

6.2.2.2 Link Evaluation

If clusters are perfect, which means that each cluster matches exactly with an annotated incident, evaluating links has a trivial solution. The same incidents can be found in both the system output and the ground truth, and a comparison of the existence for the corresponding links is straightforward. Unfortunately, the clusters are erroneous in most cases, and there is not a strict correspondence between a system cluster and an incident, so the annotated links between those incidents cannot be directly mapped to links among clusters.

To overcome the difficulty, an alternative approach can be taken, which assumes that the links exist between passages instead of incidents or clusters. If there is a link from an incident with p_i passages to another incident with p_j passages, it is equivalent to $p_i{*}p_j$ links that go between all the cross-incident passage pairs.

With that conversion, a link matrix M can be defined. If there are s passages, an $s*s$ matrix is formed, and an element in it is defined as

$$M_{ij} = \begin{cases} 1 & p_i \rightarrow p_j \\ -1 & p_j \rightarrow p_i \\ 0 & otherwise \end{cases}$$

With the definition above, two link matrices can be easily generated, one (MT) comes from the annotation, and the other (MS) from the system output. Then the pair-wise link precision and recall will be defined as simple matrix calculations:

$$P_{link} = \frac{\sum_{i,j} | MS_{ij} * MT_{ij} |}{\sum_{i,j} MS_{ij} * MS_{ij}}$$

$$R_{link} = \frac{\sum_{i,j} | MS_{ij} * MT_{ij} |}{\sum_{i,j} MT_{ij} * MT_{ij}}$$

Here the numerator calculates the number of locations where the elements in both matrices are 1 or -1, and the denominator counts the 1s and -1s in one matrix only. These evaluations are equivalent to the pair-wise link precision and recall defined in Chapters 4 and 5, although they are described in a different way. Of course, the basic elements are passages instead of stories.

In the calculation above, a special case is ignored when the element is 1 in one matrix but -1 in another. These are the passage pairs where the system identifies the correct link but marks it in the wrong direction. The proportion of arrows pointing the wrong way can also be easily calculated with these link matrices.

$$Err_{link} = (1 - \frac{\sum_{i,j} MS_{ij} * MT_{ij}}{\sum_{i,j} | MS_{ij} * MT_{ij} |}) / 2$$

6.2.2.3 Combination of Evaluation Measures

With the evaluation criteria defined above, the performance of a system can be measured in different aspects. However, it is still difficult to compare two algorithms directly, as one may achieve a better score for one aspect but lose in another. Under such a scenario, there cannot be a strict preference between those algorithms as we do not have a single number that shows the overall performance.

From the description of the evaluation measures, they are divided into two categories. One mainly describes how close the incidents from the ground truth and the system-generated clusters are. The other focuses on the quality of links.

As the incident concentration and cluster purity are often negatively correlated, an average can be taken to achieve a fair balance between them. Precision and recall in IR have a similar relation, and they are often combined by taking the harmonic mean[1] [van Rijsbergen 1979]. Likewise, concentration and purity can be merged to return an overall evaluation on the cluster quality.

$$Mean_{cluster} = \frac{2 \times concentration \times purity}{concentration + purity}$$

[1] The harmonic mean of two positive numbers is influenced more by the smaller one. It is usually used in cases that both number require high values. If one of them is large but the other is small, the harmonic mean cannot exceed twice of the smaller value.

In the evaluation of the link quality, precision and recall of links are already defined, but one additional factor must be taken into consideration. If a link is established at the correct location but placed in the opposite direction by mistake, it is not regarded as a correct link. Therefore, the links are evaluated with a slightly different average of precision and recall.

$$Mean_{link} = \frac{2 \times P_{link} \times R_{link}}{P_{link} + R_{link}}(1 - Err_{link})$$

As both the clustering part and the threading step are important for the final quality of the incident network, these two are combined to form an overall performance score of the system. Since the link quality is limited by the performance of the clustering step, the evaluation in links usually returns a lower value. Therefore an arithmetic mean is inappropriate as the larger value will shadow the smaller one. We use their harmonic mean instead:

$$Mean_{all} = \frac{2 \times Mean_{cluster} \times Mean_{link}}{Mean_{cluster} + Mean_{link}} \tag{5}$$

In the calculation, the smaller factor has more impact on the final score. This tendency is desirable in our framework, as links are usually more important in forming an incident network.

With this single-valued evaluation score, it becomes more convenient to compare various systems, and a strict training/test set division can be implemented to optimize the parameters.

6.2.3 Distance-Based Matrix Comparison

A single-valued evaluation score has been defined in the previous section, which combines five different measures with harmonic mean calculations. However, the combination is arbitrary to some extent, and the final score does not directly reflect a single physical feature of the incident network (except that it shows the trade-off between the performance in the clustering and link components). In this section, another single-valued evaluation will be introduced, which directly represents the similarity between the ground truth and the system output.

Inspired by the matrix comparison method in the link evaluation, more complex relations can also be modeled by a matrix. When two passages are randomly selected from the collection, their relation is determined by how close they are. If they look similar, it is very likely that they belong to the same incident, and a distance of 0 can be assigned. If they are not close enough but some correlation can be inferred between the passages, there may be a link between these two, and the distance is 1. If they are completely irrelevant with no possible relation, the distance should be ∞ (infinity). Such a distance also defines the proximity of different relations. When the ground truth claims that two passages are in the same incident but the system assigns a link between them, it is an error but better than declaring the passages as unrelated.

Similarly to the link matrix, a distance matrix D can be defined as:

$$D_{ij} = \begin{cases} 0 & p_i \approx p_j \\ 1 & p_i \rightarrow p_j \\ -1 & p_i \leftarrow p_j \\ \infty & p_i \circ p_j \end{cases}$$

The four different relations in the equation above are, members of the same incident (cluster), link to, link from, and unrelated, respectively.

With two distance matrices, *DT* from the annotation and *DS* from the system, the similarity between them can be calculated. The score function for the corresponding elements in the two matrices has a universal format. Since this equation is symmetric, it does not matter whether *a* is an element in *DT* or *DS*.

$$f(a,b) = \frac{1}{\| a | - | b \| + 1}$$

It is desirable to have a closed format for the score calculation, but this equation has omitted one case. It returns 1 when a and b are 1 and -1, respectively. It means that the system has found the link but in the wrong direction, so a score smaller than 1 should be assigned for that case. With this adjustment, the complete value table for the score function is shown in Table 13.

a b	0	1	-1	∞
0	1	0.5	0.5	0
1	0.5	1	0.5	0
-1	0.5	0.5	1	0
∞	0	0	0	0

Table 13: Value Table for Score Function f(a,b)

The score function will be added throughout all the locations in the distance matrix. In order to limit the final value of the evaluation measure within the [0, 1] range, it needs to be normalized with the maximal value that the sum can achieve. Unlike the asymmetric format in Section 5.4, the matrix similarity function is defined as:

$$Sim(DT, DS) = \frac{\sum_{i,j} f(DT_{ij}, DS_{ij})}{\sum_{i,j} \max(f(DT_{ij}, DT_{ij}), f(DS_{ij}, DS_{ij}))}$$

The denominator is the number of locations that the element is not ∞ in at least one of the matrices. This evaluation method is similar to the matrix comparison algorithm in relation-oriented story threading described in Chapter 5 [Feng and Allan 2007], but it is applied to passages instead of stories. Although the calculation does not involve the rich relation types in the earlier measure, it provides a symmetric evaluation (the result is the same if we swap *DT* and *DS*), while the previous matrix comparison places more of its attention on the truth annotation.

To test the robustness of this evaluation, we have considered a few ideal and degenerate cases:

1. DS is a perfect match of DT. The similarity is obviously 1.
2. The system assigns all passages to the same cluster, then DS is all 0. The numerator will be between $0.5f(DT_{ij}, DT_{ij})$ and $f(DT_{ij}, DT_{ij})$, but the denominator is *s*s* (if there are *s* passages in the whole collection), so the similarity is small.
3. The system outputs n singletons without any link, and DS is all ∞. The numerator will be 0, and the denominator is $f(DT_{ij}, DT_{ij})$, so similarity 0 is returned.
4. The system outputs n singletons that are fully linked, which makes DS all 1 or -1. It is similar to case 2.

5. The system generates perfect clusters but no link. The portion of the score where DT is 0 can be earned, but other parts are missing, and the denominator is the same as case 1 ($f(DT_{ij}, DT_{ij})$).
6. The system gets all the links correct but no cluster. Similar to case 5.

This method returns evaluation results as expected in all the cases above, so it should work well as a single-valued evaluation for our experiments. However, the evaluation results from our preliminary experiments are low (usually < 10%), which are much smaller than most evaluation measures that we have observed in IR runs (usually 30-80% F-values are observed in a retrieval experiment, depending on the complexity of data and task). Next the output of this evaluation will be analyzed with a few degradation experiments to find the reason.

Starting from an ideal incident network, which should have 100% evaluation result with the above method, noise in various types is gradually inserted into the system, and the performance drop is recorded with the increase of noise.

1. Remove a certain percentage of positive (violent) passages and replace them with approximately the same number of negative (non-violent) samples.
2. Randomly reassign cluster label for a certain percentage of positive passages.
3. For the links, randomly change the source and destination for some of them.

The evaluation results are shown in Figures 25-27. It is observed that the performance drops nonlinearly with the insertion of noise in the first two types. Only the third graph shows an approximate linear decrease, but the third type is not the main source of noise, especially for queries that do not contain many links.

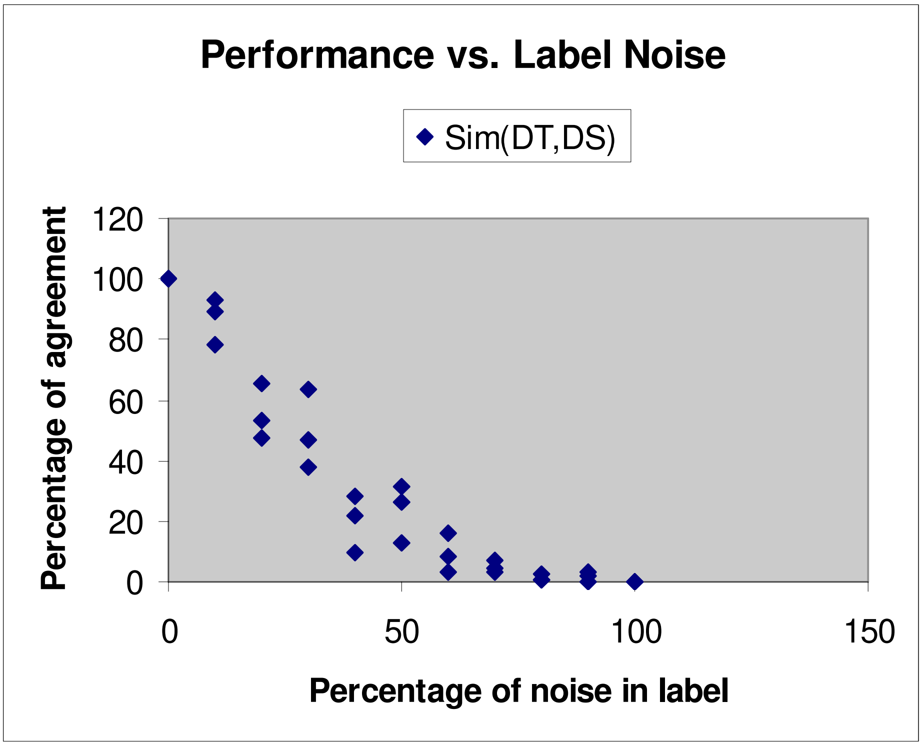


Figure 25: Evaluation Results vs. Noise in Type 1

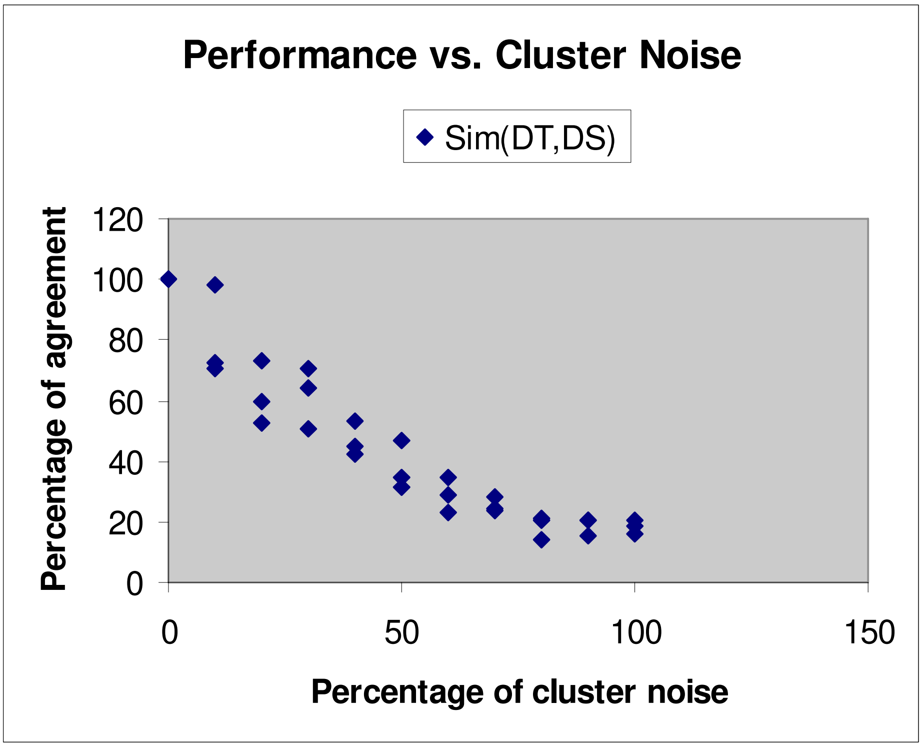


Figure 26: Evaluation Results vs. Noise in Type 2

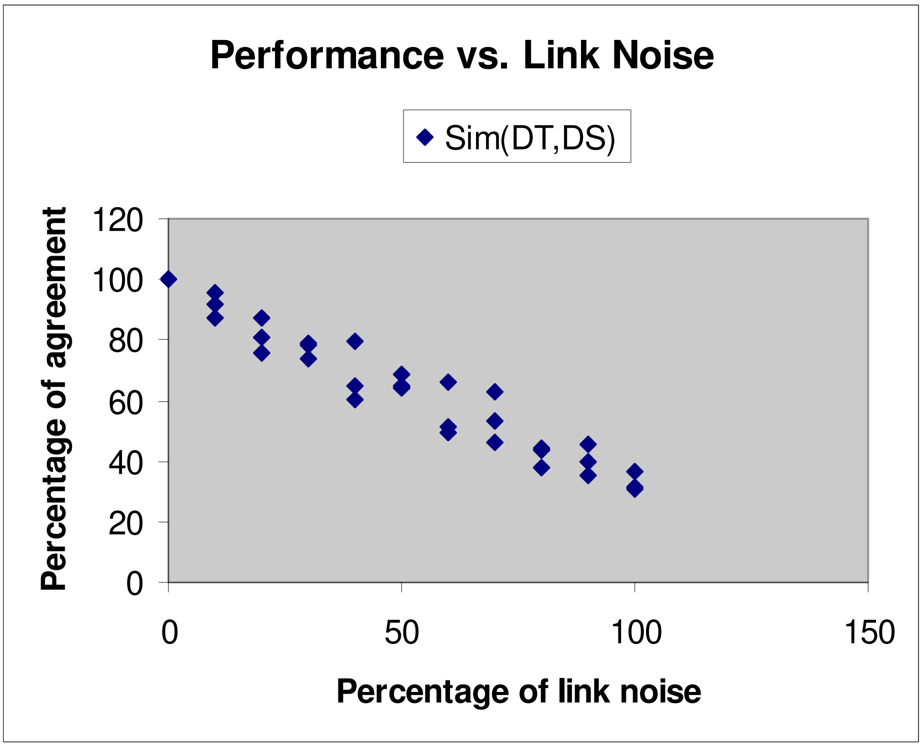


Figure 27: Evaluation Results vs. Noise in Type 3

Additional regression analysis shows that the evaluation results decreases quadratically with the introduction of the first two types of noise, which means that the square root of the matrix comparison evaluation well approximates the percentage of noise in these types. Since these are the main sources of errors in the experiment, it is advisable that this change be made in the evaluation to reflect the noise better.

$$SQ\_Sim(DT,DS) = \sqrt{Sim(DT,DS)} = \sqrt{\frac{\sum_{i,j} f(DT_{ij}, DS_{ij})}{\sum_{i,j} \max(f(DT_{ij}, DT_{ij}), f(DS_{ij}, DS_{ij}))}} \qquad (6)$$

6.3 Calibration Study

In preliminary experiments, performance based on Equation 6 ranges from 0% to 70%, which varies on the algorithm and the difficulty of the query. However, it is an open question when we can claim a system to be “good enough.” In order to explore its utility in a real application, the incident threading framework needs to go through a calibration experiment to show the performance level at which it really works.

6.3.1 Design of Study

There are many different ways to perform the study, but the basic principle is that an objective standard must be available to avoid personal preference. For instance, if an incident network is shown to the user and the question is if he/she likes it, it is inevitable to bring personal opinion into the evaluation. Each individual has his/her own standard of a “good” system, and there is no known method to force all of them to agree on a “gold” standard.

The calibration study is designed in the following way. Given an existing query, a certain number of top-ranked documents are collected. These documents are processed through an incident threading system, and the system outputs an incident network. Then an annotator is provided with one version of the information, either the original documents or an incident network, together with a list of questions that are directly related to the original query and based on the content of the documents. In a limited time, the annotator browses through the information he/she has, and tries to find as many answers as possible.

In order to find a precise objective for the performance level, multiple versions of the incident network are supplied. The original documents have no variance, but the incident networks can include different proportions of noise, which change their performance in the evaluation.

Inspired by the noise analysis in the previous section, similar actions are taken to generate various incident networks.

- Start from an incident network that 100% matches the ground truth.
- Introduce the same percentage of label, cluster and link noise (refer to the previous section for details).
- As the noise is randomly assigned, the evaluation result is not always the same even with the same percentage of noise. Run the same system 10 times, and record the matrix similarity score every time. Then calculate the average of the different runs.
- Run the noise introduction algorithm repeatedly until it returns an evaluation result close to the average of the previous runs.
- Generate the corresponding incident network and record with the evaluation score.

One may wonder why the matrix comparison evaluation is used instead of the cluster-link mean. There are two single-valued evaluation measures defined in Section 6.2. The first, which is the harmonic mean of various accuracy criteria (Equation 5), is affected mainly by the link performance, since the links are usually more difficult to be identified correctly. The second, which is a direct comparison between two matrices (Equation 6), relies more on the majority of elements in the distance matrix. For many queries, the number of 0's (passage pair with the same incident membership) is much larger than that of 1 or -1's (passage pair with a link between the corresponding incidents), so the latter is often overwhelmed by the clustering performance. Most of the questions in the study are based on facts mentioned in the news, and only a few of them focus on contextual information. Therefore, the matrix similarity score is more appropriate for this application to represent the quality of an incident network.

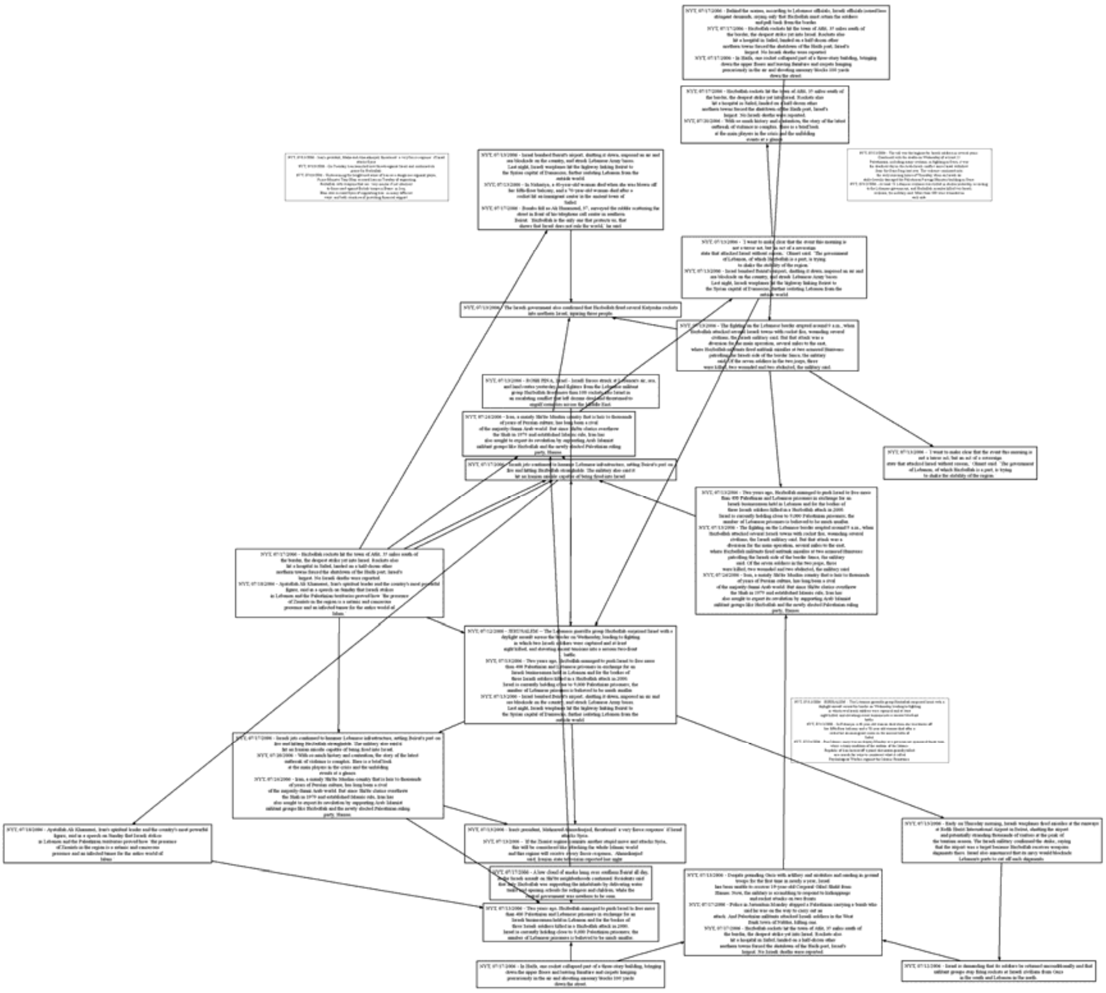
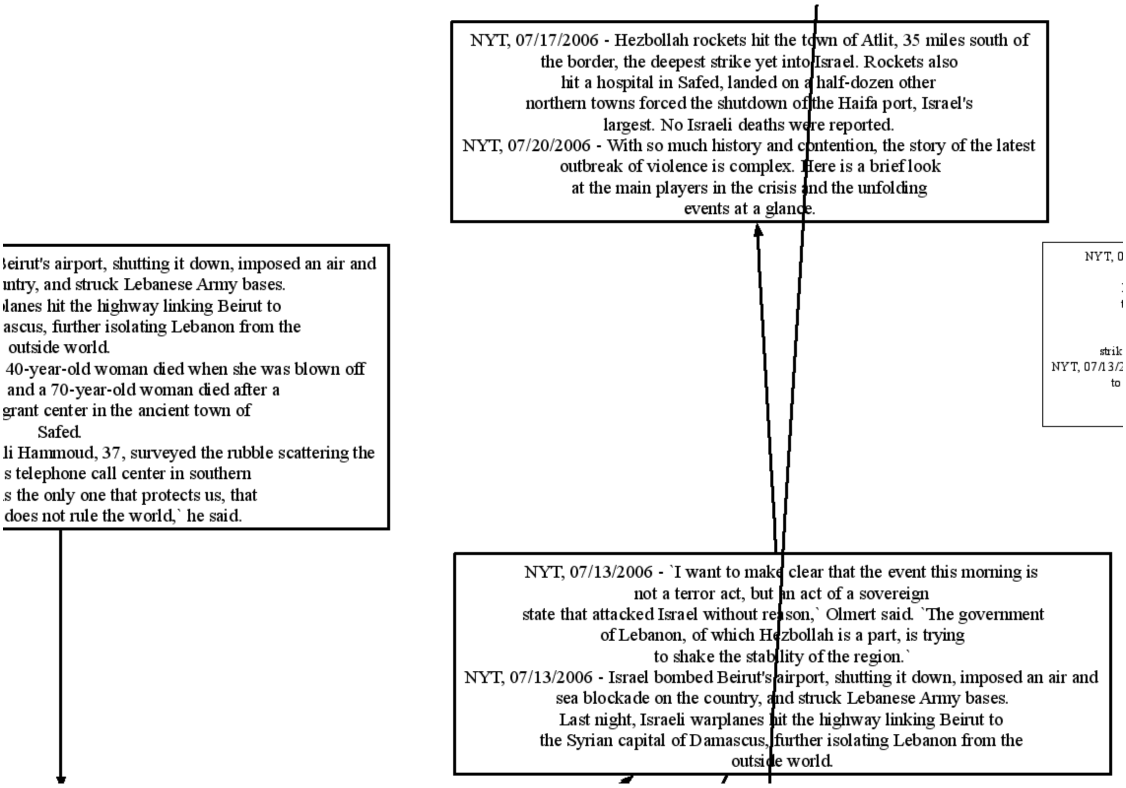


Figure 28: Sample Incident Network in Calibration Study

After generating the incident networks, multiple annotators are required. Each of them receives one version of the information, either the original documents or an incident network at a certain performance level. Their results are checked against the standard answers, and a score is assigned to each.

Figure 28 shows an example of the incident network in the calibration study and Figure 29 is a closer snapshot of it.



NYT, 07/17/2006 - Hezbollah rockets hit the town of Atlit, 35 miles south of the border, the deepest strike yet into Israel. Rockets also hit a hospital in Safed, landed on a half-dozen other northern towns forced the shutdown of the Haifa port, Israel's largest. No Israeli deaths were reported.
NYT, 07/20/2006 - With so much history and contention, the story of the latest outbreak of violence is complex. Here is a brief look at the main players in the crisis and the unfolding events at a glance.

leirut's airport, shutting it down, imposed an air and
intry, and struck Lebanese Army bases.
lanes hit the highway linking Beirut to
ascus, further isolating Lebanon from the
outside world.
40-year-old woman died when she was blown off
and a 70-year-old woman died after a
grant center in the ancient town of
Safed.
li Hammoud, 37, surveyed the rubble scattering the
s telephone call center in southern
s the only one that protects us, that
does not rule the world,` he said.

NYT, 0
strik
NYT, 07/13/2
to

NYT, 07/13/2006 - `I want to make clear that the event this morning is not a terror act, but an act of a sovereign state that attacked Israel without reason,` Olmert said. `The government of Lebanon, of which Hezbollah is a part, is trying to shake the stability of the region.`
NYT, 07/13/2006 - Israel bombed Beirut's airport, shutting it down, imposed an air and sea blockade on the country, and struck Lebanese Army bases. Last night, Israeli warplanes hit the highway linking Beirut to the Syrian capital of Damascus, further isolating Lebanon from the outside world.

Figure 29: Part of Figure 28 Enlarged

There are certain restrictions in the process of question design and reading comprehension, so that the comparison can be fair across annotators.

- The questions are mostly fact finding, where the answer can be found within a single passage. Complex reasoning is usually not required to answer the questions.
- The questions are approximately evenly distributed in the documents, without obvious tendency to ask more questions at the beginning or the end.
- The order of questions is rearranged so that it does not follow the order they appear in the documents.
- As the incident network is displayed in an image, which is usually too large to fit into a single screen, image viewing software that allows convenient zooming and scrolling is desired.

- Since it is nearly impossible to search for a keyword in a figure, search in the source documents is also prohibited.
- Reading the questions before starting the timer is allowed and encouraged, but a peek at the comprehension material is considered cheating.

Figure 30 contains 10 questions for query BAE_TR002. As we can see, they are mostly fact-finding questions, and the types include multiple-choice, yes/no judgment, brief statement and blank-filling. The answers are usually short so that the annotator can spend most of the given time on reading.

BAE_TR002: WHAT PEOPLE/ORGANIZATIONS/COUNTRIES ARE INVOLVED
IN [the Israeli-Lebanon conflict] AND WHAT ARE THEIR ROLES?

1. When did Hezbollah launch the surprise attack into Israel?
a) 1982 b) 7/12/2006 (Wed) c) 7/13/2006 (Thu) d) 7/20/2006 (Thu)

2. Did Israeli bombard the city center of Beirut, the Lebanese capital?

3. ( ) Israeli soldiers were captured in the cross-border attack.

4. What did Israel do after the attack?

5. What did Hezbollah do in response?

6. ( ) rockets were fired into Israel.

7. What kind of blockade did Israel impose on Lebanon?
a) land b) sea c) air d) two of the above e) all three of them f) none

8. In Nahariya, a 40-year-old woman died when a rockets hits ( ).
a) a balcony b) a hospital c) an immigrant center d) a bomb shelter

9. The conflict was mainly between the area of ( ) Lebanon and ( ) Israel.
a) northern northern b) northern southern c) southern northern d) southern southern

10. The total number of death in this Israeli-Lenabon conflict is ( ), if you see multiple death tolls, report the largest one.

Figure 30: Questions for Query BAE_TR002 in Calibration Study

6.3.2 Results

Four GALE queries (BAE_TR002, LDC_TR008, LDC_TR108 and LDC_TR110) are selected for the calibration study, and 10 questions are designed for each. In a query, we generate four incident networks at different performance levels. Together with the original documents, there are five representations of the information in each query. We hired five annotators, including four undergraduate students from different majors, and a graduate student in the same laboratory as the author.

In the first query (BAE_TR002: WHAT PEOPLE /ORGANIZATIONS /COUNTRIES ARE INVOLVED IN [the Israeli-Lebanon conflict] AND WHAT ARE THEIR ROLES), the annotator who works on the original documents answers 5 questions correctly out of the 10. In the four incident networks, the one with the lowest quality (16% matrix similarity with the truth) returns only 2 right answers. For the other three versions (27%, 36% and 54% match), the number of correct answers found are 6, 5 and 4, respectively.

It is worth mentioning that the design of questions for this query focuses more on the documents at the beginning, and questions appear in the same order as the source materials are in the original text. Therefore, the annotator who works on the documents has some advantage over the others. Despite the fact, it is fair to claim that incident networks show similar value for reading comprehension to the original documents, when they achieve about 30% matrix similarity.

When we come to the other queries, questions are more carefully designed so that the annotators have no obvious advantage over each other. Table 14 shows the result of the calibration study in BAE_TR002 (note the advantage of the first annotator) and three other queries.

In Table 14, the performance level of each version is the number of questions correctly answered by the annotator, and the incident networks also provide the matrix similarity scores. Because these “compressed” networks do not include all information in the original documents, an upper bound is listed for each of them, which marks how many questions can be answered given unlimited time (Note that a higher matrix similarity score does not necessarily mean better coverage of the questions, which account for only a small portion of the source documents). Items in italic fonts are incident networks that perform worse than the original documents in the study, and the underlined ones are better than the baseline (original documents).

Query	Documents	Network 1	Network 2	Network 3	Network 4
BAE_TR002	5/10	*2/5(16%)*	6/8(27%)	5/9(36%)	*4/8(54%)*
LDC_TR008	4/10	6/7(21%)	*1/5(25%)*	5/6(28%)	6/6(32%)
LDC_TR108	3/10	3/4(19%)	*2/7(26%)*	5/7(30%)	5/8(37%)
LDC_TR110	2/10	2/3(19%)	3/5(24%)	4/6(26%)	6/6(34%)

Table 14: Result of Calibration Study

As personal difference always exists among human beings, some annotators are faster than others in reading. Therefore, it is not always the case that one person does better than another when given a better representation. Nevertheless, the pattern is clear in the table, as incident networks start to become better than the original documents in the 25-30% range of matrix similarity. Unfortunately, the collection of the study is still relatively small for the purpose of drawing a strong conclusion.

To be conservative, our claim is that incident networks are at least similar to or better than the source documents in short-term reading comprehension, once they have arrived at the 25-30% range in the evaluation result. This range is set as our performance goal in the system implementation.

6.4 Baseline

Although it seems simplistic, term matching proves an easy yet effective algorithm in most IR applications. In some difficult scenarios, its performance is comparable to the more complex methods, which apply a large amount of heuristic or statistical information and therefore become susceptible to overfitting.

As agglomerative clustering and simple thresholding perform reasonably well at the story level [Nallapati et al 2004], it can be assumed that they should also work under a similar environment but with finer granularity. Here the algorithms are re-implemented to accommodate the changes of the new application, but the general process remains the same.

Similarly to the annotation process, paragraphs are indexed as the smallest semantic units in the collection, and each of them is converted into a term vector. *tf* and *idf* components are still calculated with Equations 2 and 3, but all document statistics are replaced by those at the passage level.

With a cosine similarity matrix for all passages in the collection, an agglomerative clustering algorithm is performed. The process starts with singleton clusters, where each contains exactly one passage, and the most similar cluster pair is merged in each round. When the similarity between two clusters is calculated, average link is used instead of the other alternatives (single link or complete link), since it usually generates more coherent clusters.

$$sim(C_i, C_j) = \frac{\sum_{p_k \in C_i, p_l \in C_j} sim(p_k, p_l)}{|C_i| \times |C_j|}$$

In the equation above, the numerator is the sum of all similarities of passage pairs across the boundary of two clusters C_i and C_j, and the denominator is the product of their sizes.

The agglomeration algorithm halts when the maximal similarity is lower than a preset clustering threshold. Then for all the remaining clusters, pair-wise similarities are calculated with average link, and a directed arrow is assigned to each pair with a similarity over another threshold, which is smaller than that in clustering. The arrow points from the earlier cluster to the later one, where the order is determined by the time stamp of the earliest passage in each. If their time stamps are identical, the passage that appears earlier in the news stream takes precedence.

6.5 Three-stage Algorithm

In the data annotation phase, each query needs to go through three steps. The first step tells if there exists any violent action in the current paragraph; the second annotates these actions in detail and shows their coreference; the last step creates links between the key incidents and all others. Likewise, a three-stage algorithm is implemented to simulate this process.

6.5.1 Binary Classification for Violent Passages

As observed in the annotation process, there are many passages that do not satisfy the requirement of the subject. In the annotated collection, only passages that contain the description of at least one violent action qualify for the incident selection process. It does not mean that other passages do not contain any useful information, but performance in that part does not directly affect the evaluation result shown in the previous section. In order to reduce the distraction of those data, a classification process that selects the "violent" passages will reduce the number of objects to analyze in the next steps, thus decreasing the computational cost and the percentage of noise.

This process corresponds to the first step of the human annotation, where a label is assigned to each passage whether it contains any violent actions or not. It is clearly a binary classification problem, and several classification algorithms can be tried.

Support Vector Machine (SVM) [Vapnik 1995] is a commonly-used classification algorithm [Furey et al 2000, Tong and Chang 2001], which deals with data points in high dimensional space. In a linearly separable case, two parallel hyper-planes separate the samples of two classes, and the optimal solution has the largest distance between these two planes. For most applications, there are always some samples that are placed on the wrong side, and SVM extends to a soft margin that allows such cases, while each misclassified sample is penalized based on its distance to the separation hyper plane. SVM^{light} [Joachims 2002] is a popular implementation of SVM[1], and it is used as one of the classification methods for the passages.

Maximum Entropy (MaxEnt) classifiers estimate the probability distribution from a training set with label information. "Information theory provides a constructive criterion for setting up probability distributions on the basis of partial knowledge, and leads to a type of statistical inference which is called the maximum entropy estimate. It is the least biased estimate possible on the given information; i.e., it is maximally

[1] Available at http://svmlight.joachims.org/ as of April 28, 2008.

noncommittal with regard to missing information." [Jaynes 1957] MaxEnt has been widely applied in Information Retrieval [Berger and Lafferty 1999, Jeon and Manmatha 2004]. There are many available implementations of the MaxEnt Classifier. Here a C++ toolkit is used in the experiment[1].

Boosting algorithms are also widely used for classification [Bauer and Kohavi 1999] as well as other applications [Zemel and Pitassi 2000]. Starting from simple and inaccurate assumptions, more rules are gradually appended to the model, which keeps improving the performance on the training data. While it is generally fast and able to train very accurate models for large datasets, the risk of over-fitting is also high after a certain number of rules have been added. A popular implementation of boosting that can handle plain text features - BoosTexter[1] [Schapire and Singer 2000] - is used in the classification experiment.

The features used in the classification experiments are:

- Number of terms in the passage. From our observation, a short passage usually contains no description of violent information.
- Number of terms that appear in the main characters. Presence of person or organization names often indicates that there is an incident in the passage.
- Number of terms describing locations. Geographical locations are also useful in the narrative of an incident.
- Number of terms in the time stamps. When a time point or period is mentioned, it is very likely to see a real-world occurrence.

[1] Available at http://homepages.inf.ed.ac.uk/s0450736/maxent.html as of April 28, 2008.

- Percentage of action verbs that describe violence-related events. Here action verbs are extracted with a Part-Of-Speech (POS) tagger (TnT tagger [Brants 2000]), and violent terms are manually selected from the list of action verbs.
- Percentage of terms for all variances of "be," "do," "have" and "say." These verbs tend to appear more in the non-violent passages.
- Percentage of terms that express certain extent of uncertainty, e.g., likely, may, can, often, sometimes, etc. When we see one of more of these terms, the paragraph is usually talking about an assumption or a general state. Such a passage does not include an incident.
- Combinations of the three features above.
- The full text of the passage. This feature is available only to BoosTexter, as the other two do not accept text features.

The main characteristics (e.g., characters, time, location, key verbs) of a passage are important in representing its content, where many of them belong to certain types of named entities. Named entities are useful in identifying the topic to which a news story belongs [Kumaran and Allan 2004], and Raghavan et al [2004] show that language models associated with entities can improve the performance of answering certain questions. With accurate identification of named entities, it becomes more precise to cluster duplicate news reports and track the development of a topic [Steinberger et al 2005]. In order to achieve good performance in the clustering and threading steps, these features need to be extracted with high accuracy.

[1] Available at http://www.cs.princeton.edu/~schapire/boostexter.html as of April 28, 2008.

Table 15 shows the correspondence between some features above and the objects detected in Automatic Content Extraction (ACE). Scores in the table come from the official evaluation of ACE-2005 [ACE 2005], and only the results for English collections are listed. The formulae for evaluation can be found in the evaluation plan [ACE 2008], but the score does not correspond directly to commonly-used IR evaluation measures like precision or recall. On the other hand, the approximate correlation can be estimated from published reports of individual participants. Ji and Grishman [2005] show approximately 90% F-value for the entity detection task in Chinese collections, and the same system receives a score of 65.7 in the official evaluation [Doddington et al 2004]. Although there is no strict mapping between the ACE score and F-value, they have a positive correlation. Usually a state-of-the-art retrieval system can achieve over 80% F-value for a collection with medium difficulty, so ACE scores over 60 (approximately 80-85% F-value) can plausibly be regarded as good.

Feature	Alias	ACE object	Types	Highest score	Comment
Main character	WHO	Entity	Person, organization	71.9 for all entities	Some entity types do not belong to characters but are also useful, like vehicle and weapon
Time stamp	WHEN	Time	All	63.7 for time	
Location	WHERE	Entity	GPE, location	71.9 for all entities	Some facilities also contain location information
Action	WHAT	Event	All	14.4 for events	The eight types in events cover only a small portion of actions

Table 15: Main Features of a News Passage, Categories They Belong to in ACE, and Corresponding Accuracy of Best System in ACE-2005

Table 16 shows the average classification error rates of the three algorithms. When the same set of features is used, all three achieve similar accuracy. As BoosTexter is the only one that accepts full text, it yields better performance with the additional feature. In the leave-one-out classification experiment with all 17 queries, its average error rate is 14.43%, which is significantly better than the other two in a one-tailed t-test.

Algorithm	Text feature	Average error rate	P-value in t-test
BoosTexter	Yes	14.43%	-
SVM^{light}	No	17.47%	0.0147*
MaxEnt	No	19.60%	1.36×10^{-4}*

Table 16: Performance Comparison of Three Binary Classifiers

However, failure analysis shows that the default threshold (0) for BoosTexter generates high miss rates for the positive samples, as the objective function is to minimize the number of classification errors. Since the percentage of label noise greatly affects the quality of the final incident network (refer to Figure 25), sometimes a lower threshold would improve the evaluation result since it decreases the number of misses, at the cost of increasing false alarms.

6.5.2 Incident Formation

Although passages are much shorter than news stories, text snippets that belong to the same incident must have some overlap, either in terms or in semantics. Therefore, the main algorithm used in forming incidents is clustering [Hartigan 1975, Jain and Dubes 1988] based on the similarity. Since the number of passages is manageable for each query, especially after the filtering step, an agglomerative process is adopted with slight changes in the merging rule. The overlap may be the identical terms used in both passages, a reference to the same person or organization, or a mention of the same geographical location, etc. All these are possible evidence to assign two passages to the same incident, but how to convert them into an overall probability of confidence directly affects the performance.

In the earlier work that utilizes multiple features [Feng and Allan 2007], a weighted sum of similarities from various features is calculated to determine the resemblance between two stories. It works better than a single feature (term vector), but the weight assignment among features is heuristic. When the same approach is applied to passages, the scores become more unstable, as all features are shorter. A good analogy of term vectors can be introduced by the overlap of a few terms, when they are general words instead of keywords, and then the high similarity of term vectors is enough to claim two passages as belonging to the same incident. Similar phenomena are also observed from other features, and they cause many false positives (assigning two passages to the same incident when they are not describing the same occurrence). Referring back to the definition of an incident, it involves all different attributes of the description, including main characters, locations, the time stamp and the actual happening. So here a stricter requirement is enforced that matches in all attributes must be achieved for two passages to be declared similar. This method is effectively in reducing false positives, but at the cost of missing passage pairs that belong to the same incident but mismatch in at least one aspect.

These are the main features in the clustering process:

- Similarity of all terms. As usual, it is the cosine similarity of the tf-idf term vector which represents a passage, and describes how similar two passages are based on the vocabulary in them.

- Similarity of main characters. Here main characters include both persons and organizations, and they are extracted by the Proteus system from New York University.
- Similarity of geographical locations, which includes all LOC (location), GPE (geo-political entity) and FAC (facility) entities in the ACE specification [ACE 2008].
- Match between time stamps. Here the time stamp refers mainly to an activity date mentioned in the text, as the publication dates always overlap for two passages from the same news story. If two passages both have at least one time stamp but none of them matches, a small score (we arbitrarily select 0.5, which works well in the training phase) is assigned to this feature. If at least one of them does not contain any time stamp, the pair receives a score slightly smaller than 1 (0.9 in our experiment from empirical data). If at least one time stamp matches, score 1 is returned. As many passages are missing time stamps, and a discrete feature (it can only be 0.5, 0.9 or 1) is inappropriate for this application, we combine this feature with the term similarity by multiplying them.

A threshold for each of these features is assigned, and two passages are merged into the same incident only when their similarity in each aspect satisfies the corresponding threshold. The only exception is the last feature, which is merged with the term similarity and does not require a separate threshold. Many short passages miss the field of geographical location if it has already been mentioned in the context, so the threshold for that feature is usually low.

6.5.3 Contextual Links

As the assignment of relation type is not mandatory in this implementation, no attempt is made to distinguish one type of link from another. Analysis shows that most contextual links in the violence subject belong either to *consequence* or *reaction* in the logical category, or *follow-up* in the progressional form. Links in these types usually contain two incidents, which happen at different yet close times, involve the same geographical location, mention similar main characters, but often show poor term overlap.

With those observations, the same set of features as in the previous step is used. Preliminary experiments show that the threshold of term similarity is usually much lower than the corresponding value in clustering. On the contrary, thresholds for main characters and location overlap are often higher.

The strategy to determine the link direction in previous experiments does not work well in this algorithm, because the number of links in the wrong direction often exceeds 50%. As the publication dates for all passages in one story are always the same, there are a large number of ties if the earliest passage in each incident represents its time stamp, and many mistakes are committed when the direction is randomly assigned or enforced by the order in news stream. An alternative method is to compare the time order of all passage pairs across the boundary of two incidents, and the side that receives more precedence votes is regarded earlier. There are still errors caused by this assignment, but overall it is a better rule than the earlier one.

6.6 Experiments

A formal training/test division is necessary to justify the experiment results, as more complex models usually have an advantage in achieving higher performance on the training set. At the same time, more heuristic information and more parameters also increase the risk of overfitting, which will hurt the evaluation result on the test set.

Unfortunately, the passage-based experiment does not have a large data collection with relevance judgment, which limits the scope of training. If the training set accounts for a large percentage of the corpus, the test set would be too small to infer meaningful conclusions, especially for a significance test. Under such a condition, cross validation is a good choice. Since the number of queries is small, leave-one-out cross validation is performed, where the data in one query are reserved for evaluation in each round and all others can be used for training.

There are only two parameters to tune for the baseline algorithm.

1. The clustering threshold, where the agglomerative process halts when the maximal similarity drops below it.
2. The link threshold, which is used to decide if a link should be created between an incident pair.

As the three-stage algorithm involves more features, the number of parameters is also larger.

1. The filtering threshold for BoosTexter. The default is 0, but more passages are allowed in later steps when it is set at a lower value.
2. The clustering thresholds for term vectors (merged with time stamp information, see the discussion in Section 6.5.2), named entities and locations.

3. The link threshold for term vectors (with time), named entities and locations.

In the training phase, each parameter is swept through its proper range with a small step, and the parameter setting that optimizes the performance in the training set is kept. The objective is to maximize one of the single-valued evaluation measures – either the harmonic mean of various scores (Equation 5) or the matrix comparison measure (Equation 6).

When the number of parameters is large, the search space grows exponentially, making it intractable to calculate the performance for each parameter combination. In the implementation of parameter tuning, one parameter is optimized within its range in each round, while others are fixed. This process continues until the performance does not improve with any change of a single parameter. Note that this search method is possible to be trapped in a local maximum, but generally it returns a good configuration.

Evaluation	Baseline	Three-stage	Change in %
Incident concentration	0.1985	0.2609	+31.4%
Cluster agreement	0.1494	0.2703	+80.8%*
Clustering precision	0.1427	0.2830	+98.3%*
Clustering recall	0.1445	0.2161	+49.4%*
Link precision	0.0345	0.1598	+362.5%*
Link recall	0.1574	0.1866	+18.5%
Link direction error	0.3995	0.4295	+7.4% (worse)
$Mean_{all}$	0.0361	0.0654	+80.1%*
SQ_SIM(DT,DS)	19.10%	26.40%	+38.2%*

Table 17: Performance Comparison for Passage-based Systems – $Mean_{all}$ Optimized

Two sets of parameter tuning are performed on the training set, where different evaluation criteria are optimized. When the harmonic mean in Equation 5 is used, the performance data on the test set is shown in Table 17. Table 18 contains similar data, but the matrix comparison score in Equation 6 is optimized instead. Changes with an asterisk are significant improvements by a one-tailed t-test. Note that smaller numbers are better for the link direction error.

Evaluation	Baseline	Three-stage	Change in %
Incident concentration	0.3099	0.3864	+24.6%
Cluster agreement	0.1073	0.1855	+72.9%*
Clustering precision	0.1146	0.1807	+57.6%*
Clustering recall	0.2691	0.3472	+29.0%
Link precision	0.0380	0.0350	-7.8%
Link recall	0.0226	0.0113	-49.8%
Link direction error	0.2166	0.2678	+23.6% (worse)
$Mean_{all}$	0.0133	0.0110	-17.8%
SQ_SIM(DT,DS)	22.58%	25.05%	+10.9%

Table 18: Performance Comparison for Passage-based Systems – *SQ_SIM(DT,DS)* Optimized

People may notice that $Mean_{all}$ does not equate with the calculation of Equation (5) from the measures above it. All values in Tables 17 and 18 are averaged on the 17 queries in the experiment corpus, but the harmonic means are calculated on a per-query basis. Therefore, the arithmetic average in the tables may hide the unequal distribution between different queries, but $Mean_{all}$ is small for a query where one of the evaluation measures is bad. That is why the cluster-link mean always looks lower than the values above.

If we compare the results in Tables 17 and 18 to the story-based experiments in the previous chapters, the corresponding evaluation measures are smaller. This decrease of accuracy is within expectation, as news processing with passages is more difficult than that with complete stories, which contain complete contextual information and describe clearer ideas. Even with the great challenge, the three-stage algorithm performs reasonably well in comparison to the baseline.

With different measures to optimize, the two systems show interesting performance patterns. In Table 17, the harmonic mean of various scores is the objective for the parameter tuning, which focuses on the quality of both clustering and links. As the threading step is often the bottleneck of performance, moderate numbers are shown for both systems, but the three-stage algorithm is comparably more successful. For both single-valued evaluation measures, the three-stage algorithm is significantly better than the baseline. However, there is a large proportion of links that go in the wrong direction in three-stage, and it does not receive a high score on the recall part. From our failure analysis, the reason is that many positive passages are erroneously filtered out in the first step.

When the matrix comparison measure is used to optimize the parameters (Table 18), the performance difference between these two systems becomes more complex. As this evaluation algorithm favors pair-wise relations that dominate the distance matrices, clustering performance is weighted more than links, because the number of pair-wise connections is small for most queries. Under that condition, the three-stage algorithm outperforms the baseline in clustering and the overall matrix comparison, but does worse in links, which also leads to a smaller cluster-link mean.

What evaluation criterion to use highly depends on the application. For fact-finding scenarios, the matrix comparison measure seems to be a better option. The calibration study in Section 6.3 is such an application. We are glad to see that the performance of the three-stage algorithm falls in the 25-30% range, which implies that the output incident network is “useful” in comparison to the original documents. On the other hand, general news representation should adopt the harmonic mean, as contextual information is a very important factor in facilitating the understanding of a large collection of news reports. More experiments would help to understand the correlation and difference between these evaluation measures.

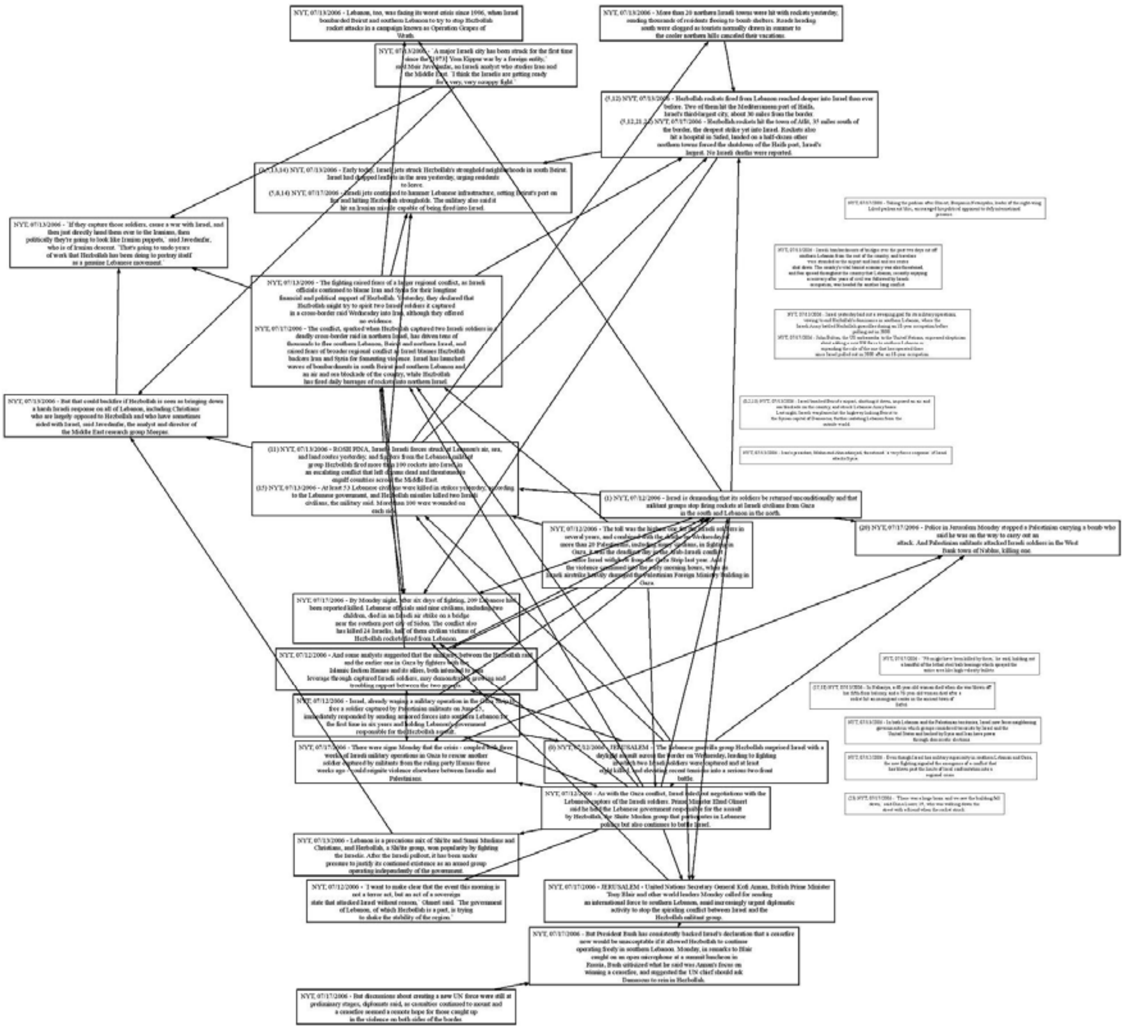


Figure 31: Incident Network of the Israeli-Lebanon Conflict

For the three documents in the Israeli-Lebanon conflict topic, the incident network generated by the three-stage algorithm is in Figure 31. Evaluation results show 0.1098 cluster-link mean and 21.49% in matrix comparison. A closer snapshot is shown in Figure 32. Numbers in the parentheses at the beginning of a passage indicate the incidents to which the passage belongs in the truth annotation. For a passage that is identified as not containing any violent action, there is no parenthesis in front of it.

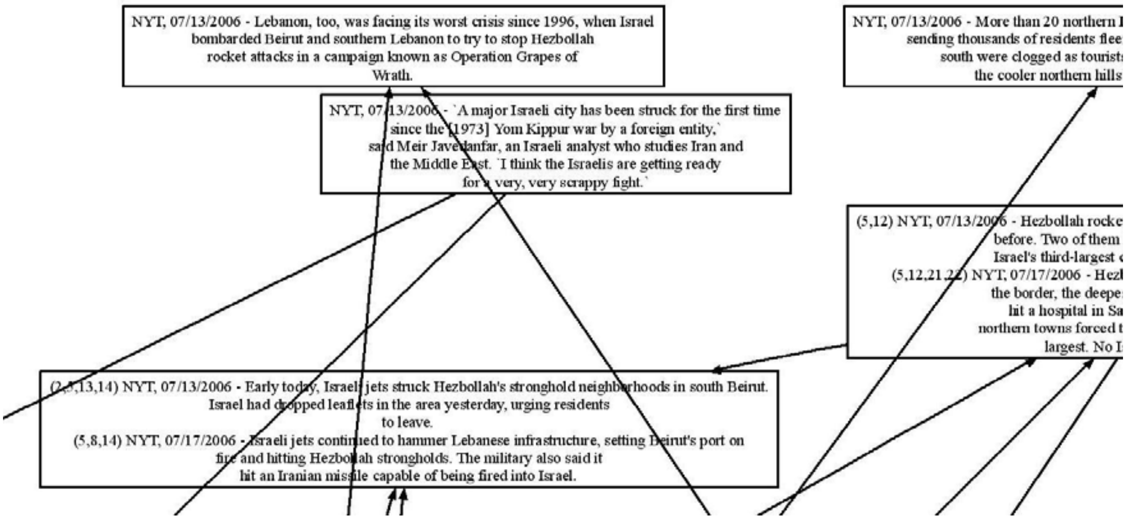

NYT, 07/13/2006 - Lebanon, too, was facing its worst crisis since 1996, when Israel bombarded Beirut and southern Lebanon to try to stop Hezbollah rocket attacks in a campaign known as Operation Grapes of Wrath.
NYT, 07/13/2006 - More than 20 northern I
sending thousands of residents flee
south were clogged as tourist
the cooler northern hills
NYT, 07/13/2006 - `A major Israeli city has been struck for the first time since the [1973] Yom Kippur war by a foreign entity,` said Meir Javedanfar, an Israeli analyst who studies Iran and the Middle East. `I think the Israelis are getting ready for a very, very scrappy fight.`
(5,12) NYT, 07/13/2006 - Hezbollah rocke
before. Two of them
Israel's third-largest
(5,12,21,22) NYT, 07/17/2006 - Hez
the border, the deepe
hit a hospital in Sa
northern towns forced t
largest. No I
(2,5,13,14) NYT, 07/13/2006 - Early today, Israeli jets struck Hezbollah's stronghold neighborhoods in south Beirut. Israel had dropped leaflets in the area yesterday, urging residents to leave.
(5,8,14) NYT, 07/17/2006 - Israeli jets continued to hammer Lebanese infrastructure, setting Beirut's port on fire and hitting Hezbollah strongholds. The military also said it hit an Iranian missile capable of being fired into Israel.

Figure 32: Part of Figure 31 Enlarged

6.7 Summary

In this chapter, the earliest work to implement an incident threading system at the passage level is discussed, mainly on five aspects – data collection and annotation, evaluation, performance goal, algorithms, and experiments.

As there is no existing text collection with snippet-incident judgment, certain data from the GALE corpora are selected for a specific class of queries. Then the annotation process is described in details, where the first step is to identify paragraphs with violent actions in them, the second marks incidents in those paragraphs, and the last step builds the contextual link. A good inter-annotator agreement is achieved, which justifies the selection of the topic of interest.

The traditional evaluation methods do not directly apply to this novel application, so special measures should be designed to evaluate the performance of certain implementations. The evaluation section starts with an analysis of attributes for a good evaluation, introduces several evaluation algorithms that focus on different aspects of the incident network, and next combines them with harmonic mean. As another single-score criterion, a matrix comparison evaluation method is proposed, and it is further analyzed through test cases and degradation experiments.

Since the evaluation methods do not provide a degree of “usefulness,” a calibration study verifies the value of incident threading for news comprehension. Although the scale of the study is small, it suggests that, for reading comprehension in a short time, an incident network with 25-30% performance in the matrix comparison measure is as good as, or sometimes better than, the source documents, which are too long to understand quickly.

Finally, we introduce two different implementations of the passage-based system. The baseline is borrowed from previous work, while the stories are replaced by passages. Another algorithm, which is composed of three stages, simulates the annotation process. The goal of the first step is to remove the passages that do not contain interesting information; the second step forms the incidents through a clustering process; and the last stage creates links based on the same set of features in the previous step, but with different requirements. Experiments show significant improvements when cluster-link mean is optimized, but only clustering performance is increased when we tune on the matrix similarity measure. Evaluation results in the experiments satisfy the goal set by the calibration study, so the incident networks are “useful” at the current performance level.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In order to stay up to date, it is important to keep track of the newest reports at any time. However, the rapid growth of information demands external aid, otherwise users may be easily overwhelmed by the huge amount of news. As each of the existing automatic news service has built-in deficiencies, this thesis describes a new framework – incident threading, which analyzes news based on the real-world occurrences discussed in a report and identifies contextual information among news incidents.

Two story-based implementations of incident threading are presented first. The earlier story threading model introduces the internal analysis of a news topic, and the later relation-oriented work extends its infrastructure by bringing link types into the contextual relations. Both of them show successful results in experiments, with an assumption that each news story describes at most one incident.

As a further extension to the previous work, we describe passage threading which breaks each news story into finer granules. This is a research area that has not been extensively studied before, so it possesses both great potential and difficult challenges. The implementation starts from a fully-annotated data collection and appropriate evaluation measures for the new application. Two algorithms are provided for a good reference of the performance. Due to the focus of the evaluation, limitations of the algorithm itself and the small collection size, the three-stage algorithm achieves significant improvement over the baseline when we tune on cluster-link mean, but less improvement is observed when matrix comparison is used in training. The calibration study shows that the current performance of an incident network is comparable to the original documents. Therefore, the application of incident threading is justifiable in a real system.

As an early attempt in the new research area, the thesis has provided a detailed framework and sufficient support for additional development. The current progress is encouraging, and further research in this direction is promising. This work has made contributions on both theoretical and technical aspects.

In the research area of automatic news processing, TDT places its entire emphasis on how to manage the correct assignment of stories into topics and does not consider the organization of an individual news topic. Story threading [Nallapati et al 2004] starts automatic analysis of the internal structure of topics, and it implements a systematic organization of news events, although the concept has been available from the beginning of TDT.

Correlations or links between news events are discussed in story threading, but its implementation is oversimplified by the binary assumption. Relation-oriented story threading [Feng and Allan 2007] defines links in specific types - logical, progressional or weak, which brings more insights into the contextual analysis of related incidents. It also introduces the global optimization framework, which generates clusters in similar quality but returns much better results on links, as displayed by the experiment.

With enough successes in the story-level news analysis, this thesis attempts to find events/incidents at a smaller granularity, which breaks the long-existing assumption in TDT that each story talks about only one occurrence. Since no data collection is available with incident annotation on snippets, it creates the earliest available corpora with incident markup. When none of the existing evaluation measures directly applies to this novel application, new approaches are designed, where some represent different aspects of the incident network, and a matrix comparison measure provides a global score for the model optimization purpose.

On the performance side, we achieve significant improvements over the baseline in both experiments at the story level, where huge increase of link quality (over 200%) is demonstrated with the successful application of a global optimization framework. At the passage level, we have observed significant increase (P-value < 5% in one-tailed t-test) in the evaluation result of one experiment but not in the other, which shows improvement over the baseline on clustering but performs worse on links.

To justify the application of incident threading in a real system, a calibration study is conducted. It demonstrates that an incident network with current quality level performs as well as or better than the original documents for a reading comprehension task within limited time. We believe that further improvement in performance will bring larger difference.

Overall, the work in incident threading has accomplished several successful implementations, but there are still many aspects for possible improvements and further extension. One of the most important contributions of this thesis is that it has established a complete framework for the promising research topic, and it will facilitate other researchers who are interested in continuing on this direction.

Currently the main challenge still lies in the proper representation of a short text snippet. Although term vectors, together with the automatically extracted main characters, geographical locations and time stamps, have been the foundation of a system with moderate performance, further research will probably need to rely on the accurate modeling of semantic information represented in the short piece of text.

Another bottleneck of the current system is that many errors are made in the classification step, and the missed positive samples greatly hurt the final performance. Currently most of the features are independent of the semantics, with few involving part-of-speech information. An ideal classifier needs to model the main topic explicitly, instead of an indirect representation from the surface features.

With the limitation of a single main subject, the possible types of relations are greatly restricted in the current implementation. An expansion to general news would be ideal, although an annotation attempt for that case failed for lack of agreement (see Appendix A). We believe that clearer instructions and extensive training should improve the inter-annotator agreement, making it possible to perform annotation for general incidents. With a richer background, type-specific relation analysis is an expectable consequence, and it will certainly help the comprehension of news evolution at a higher level.

Due to the limitation of time, resource and abilities of the author, the exploration in this thesis has to pause at this point. But with the constant increase of user need, research in this area will become more compelling. Further improvement in performance, theoretical extensions of the framework, and applications of incident threading in publicly available systems are foreseeable in the near future. These are also the aspirations of the author.

APPENDIX A

ANNOTATION AND EXPERIMENTS FOR GENERAL QUERIES

The annotation process for violent actions has been described in the thesis (Chapter 6), followed by experiments on the marked corpora. In addition to those specific queries, annotation and experiments for the general queries have also been conducted. However, its low inter-annotator agreement undermines the value of the annotation, and the binary classification experiment yields poor performance. Therefore, work in this direction is suspended. If research of incident analysis in a global domain is continued, more detailed instructions and exhaustive training will be necessary to minimize confusion in the specification of an incident.

A.1 Annotation of General Incidents

Instead of asking the annotator to mark the description of violent actions only, everything that qualifies as an incident is required to be identified for the general case. Figure 33 shows the instructions for an annotator to determine the existence of an incident. While revisiting these instructions, obvious flexibility in the judgment of an incident can be observed. For example, some incidents exist but are unimportant or irrelevant to the topic, and the instruction gives the annotation a chance to make his/her decision based on personal preference. Besides the instructions, a toy collection is provided for practice, and correct answers are included.

We define an *incident* as **a real-world occurrence that involves certain main characters, happening at specific time and location.**

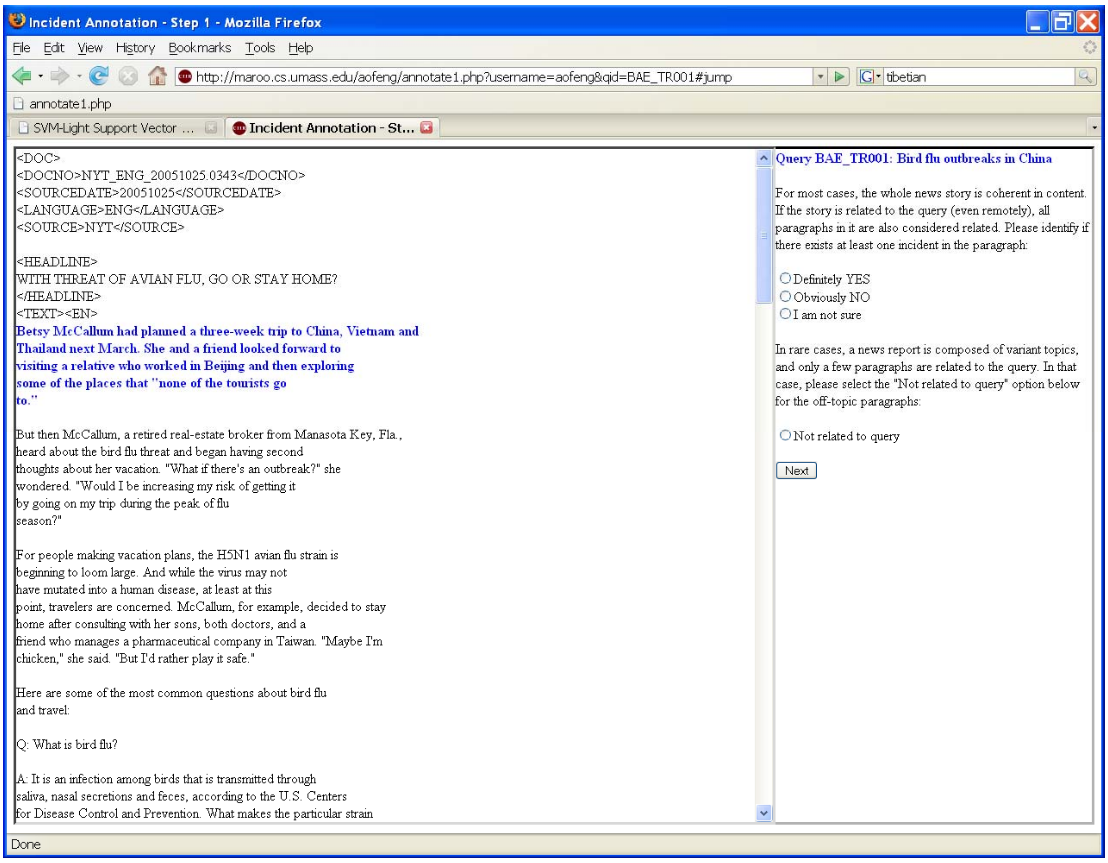
From the definition you can see that an incident has four main features: main characters (who), time (when), location (where) and occurrence (what). An incident should always contain these features, although for many cases some features are not explicitly available in the news report. “Bird flu is a disease that may infect human beings” is not an incident since it is a general description of a disease and does not involve any actual happening. “A bomb exploded today” is an incident, even if we do not see the location in the description. We know it must have exploded somewhere, and often the location can be inferred from the context.

When you are looking at a paragraph to decide on the incidents it lists, you may ignore things that are incidents but seem (to you) unimportant in the context of the news story (e.g., “the door was painted gold”). You are certainly welcome to list them all, but we do not mind if minor incidents are lost. If some part of a news report is obviously irrelevant to the topic (query), you can simply ignore everything in that part, even if incidents are mentioned. For example, a news story lists the headlines of the important reports on that day, and only one or two are related to the topic we are annotating, it is preferred to skip everything else because annotating them will take a lot of time and provide no useful information.

You are likely to encounter situations where you could generate incidents at different granularities (different levels of detail). For example, “there were bird flu outbreaks in several countries of Southeast Asia last month, including Vietnam, Thailand, Singapore and Malaysia, etc.” can be treated as multiple incidents, where the outbreak in each country is a different one, or it could be a single big incident that includes the outbreaks in the Southeast Asia area. Unless there is strong evidence to select the first option (there have been incidents that talk about outbreaks in Vietnam, and outbreaks in Singapore), always go with the second choice.

Figure 33: Annotation Instructions for General Incident

The annotation process is similar to the specific case, but the first step is slightly different. In addition to two options of yes and no, the annotator is allowed to say “I’m not sure” for obscure paragraphs. Since the number of general incidents is usually large within the top 10 documents, another option is provided to filter out descriptions that qualify as incidents but are obviously off-topic. Figure 34 shows the annotation interface of the first step.


Incident Annotation - Step 1 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://maroo.cs.umass.edu/aofeng/annotate1.php?username=aofeng&qid=BAE_TR001#jump
tibetian
annotate1.php
SVM-Light Support Vector ...
Incident Annotation - St...
Next
Done

```
<DOC>
<DOCNO>NYT_ENG_20051025.0343</DOCNO>
<SOURCEDATE>20051025</SOURCEDATE>
<LANGUAGE>ENG</LANGUAGE>
<SOURCE>NYT</SOURCE>

<HEADLINE>
WITH THREAT OF AVIAN FLU, GO OR STAY HOME?
</HEADLINE>
<TEXT><EN>
Betsy McCallum had planned a three-week trip to China, Vietnam and
Thailand next March. She and a friend looked forward to
visiting a relative who worked in Beijing and then exploring
some of the places that "none of the tourists go
to."

But then McCallum, a retired real-estate broker from Manasota Key, Fla.,
heard about the bird flu threat and began having second
thoughts about her vacation. "What if there's an outbreak?" she
wondered. "Would I be increasing my risk of getting it
by going on my trip during the peak of flu
season?"

For people making vacation plans, the H5N1 avian flu strain is
beginning to loom large. And while the virus may not
have mutated into a human disease, at least at this
point, travelers are concerned. McCallum, for example, decided to stay
home after consulting with her sons, both doctors, and a
friend who manages a pharmaceutical company in Taiwan. "Maybe I'm
chicken," she said. "But I'd rather play it safe."

Here are some of the most common questions about bird flu
and travel:

Q: What is bird flu?

A: It is an infection among birds that is transmitted through
saliva, nasal secretions and feces, according to the U.S. Centers
for Disease Control and Prevention. What makes the particular strain
```

Query BAE_TR001: Bird flu outbreaks in China

For most cases, the whole news story is coherent in content. If the story is related to the query (even remotely), all paragraphs in it are also considered related. Please identify if there exists at least one incident in the paragraph:

○ Definitely YES
○ Obviously NO
○ I am not sure

In rare cases, a news report is composed of variant topics, and only a few paragraphs are related to the query. In that case, please select the "Not related to query" option below for the off-topic paragraphs:

○ Not related to query

Figure 34: Annotation Interface of Step 1 for General Incidents

The annotation for the next two steps is identical to the violent action case, so they are not displayed here.

A.2 Inter-annotator Agreement

Following the inter-annotator agreement calculation in Section 6.1.3, the same agreement measures are applied to the general incidents. The Fleiss' Kappa is

$$P_a = 0.423$$
$$P_e = 0.285$$
$$\kappa_F = 0.193$$

for a collection of four queries, each judged by four different annotators. The pair-wise agreement is still represented by Cohen's Kappa (Table 19):

Cohen's Kappa	Annotator 1	Annotator 2	Annotator 3	Annotator 4	Average
Annotator 1	-	0.105	0.296	0.445	0.282
Annotator 2	0.105	-	0.153	0.116	0.125
Annotator 3	0.296	0.153	-	0.227	0.225
Annotator 4	0.445	0.116	0.227	-	0.263

Table 19: Pair-wise Annotator Agreement for General Incidents

The type-specific accuracy is:

P (B says yes | A says yes) = 0.540

P (B says no | A says no) = 0.404

P (B says not sure | A says not sure) = 0.153

P (B says off-topic | B says off-topic) = 0.440

It is obvious that the introduction of more labels has greatly decreased the inter-annotator agreement. To exclude the influence of the additional labels, another set of agreement is calculated, where any passage marked "not sure" or "off-topic" by at least one annotator is skipped. With this change, the new Fleiss' Kappa is

$$P_a = 0.662$$
$$P_e = 0.511$$
$$\kappa_F = 0.309$$

It is considerably higher than the previous result, but still not enough to claim good inter-annotator agreement.

A.3 Classification of General Incidents

In order to verify if passages with incidents can be distinguished from those without, a classification experiment is conducted. For the classification algorithm, the same three – SVM, MaxEnt and BoosTexter – are used. The features in the classification experiment are similar to those for the specific incidents.

- Number of terms in the full text, main characters, time stamps and geographical locations.
- Number of action verbs. They appear as tags VB, VBD, VBN and VBP in a POS tagger.
- Number of terms that are various forms of *be*, *do*, *have* or *say*.
- Number of terms that are often observed in non-incident passages, including *must*, *normally*, *because*, *may*, *not*, etc.
- Combinations of features above.

With different feature combinations and parameter settings, there are slight variations in the performance. However, the three classification algorithms consistently yield error rates around 40% in a leave-one-out cross validation of four annotated queries, which is obviously too high for an application. Therefore, experiments in the later steps are not performed.

A.4 Summary

An incident has been formally defined in Chapter 3, but its underlying implication in a general application is still obscure. Despite the effort with detailed instructions and a practice case to train the annotators, their judgments are highly subjective and fail the test of inter-annotator agreement. Even with the confusing test cases removed, they cannot achieve a consensus. Classification experiments further prove that the difference between general incidents and non-incidents is not clear, and the current performance on general incidents is not sufficient to be applied to an automatic news processing system.

Although the specific queries are valuable in the news analysis of certain subjects, they cannot provide sufficient coverage for the broad range of news. Effective processing of general news information is the ultimate goal, but a clearer description, more detailed instructions and in-depth training are required to arrive at the destination.

APPENDIX B

SIMULATED ANNEALING

The process of the simulated annealing algorithm is:

```
1. Initialize relation matrix $R$ as all -1, except for the diagonal elements that are 0. Calculate initial score $S$. Set initial temperature $T$=1000.
2. Record current best solution $RB=R$, $SB=S$.
3. While ($T$>0.01)
        (a) Save current state $R_0=R$. Calculate score $S$.
        (b) Randomly select a text snippet $i$.
        (c) Select another snippet $j$ according to the distribution of $R_{i*}$.
        (d) Change the value of $R_{ij}$, update corresponding elements in matrix $R$ to keep the restrictions satisfied.
                i. $0 \rightarrow -1$: break a cluster into two.
                ii. $0 \rightarrow +$: break a cluster into two, and select the relation $R_{ij}$ that maximizes $Rule(p_{i,}p_j, R_{ij})$.
                iii. $-1 \rightarrow 0$: merge two clusters.
                iv. $-1 \rightarrow +$: build a link with relation $R_{ij}$ that maximizes $Rule(p_{i,}p_j, R_{ij})$.
                v. $+ \rightarrow -1$: disconnect a link.
                vi. $+ \rightarrow 0$: merge two clusters.
        (e) Calculate new score $SN$.
        (f) If ($SN$>$SB$) $RB=R$, $SB=SN$.
```

(g) If $random(0,1) < e^{\frac{SN-S}{T}}$, keep the change of *R*. Otherwise, *R*=*R₀*.

(h) *T*=*T**0.99.

4. Return the best solution *RB*.

APPENDIX C

EVALUATION MEASURES

In the three implementations of incident threading, some evaluation methods have been defined in each. Some of these criteria measure the same or similar attributes of a system, but are described in different format. In order to provide a clearer view of their correlation and difference, we list all the evaluation measures that have been mentioned in this thesis in Table 20.

Evaluation measure	Representation	Explanation	Section number(s)
Clustering precision	CP, $P_{cluster}$	Proportion of correct story (passage) pairs in system output	4.3, 5.4, 6.2.1.1
Clustering recall	CR, $R_{cluster}$	Proportion of story (passage) pairs found in ground truth	4.3, 5.4, 6.2.2.1
Clustering F-value	CF, $F_{cluster}$	Harmonic mean of *CP* and *CR*	4.4, 5.4
Binary link precision	DP, P_{bin}, P_{link}	Proportion of correct story (passage) pair-wise links in system output	4.3, 5.4, 6.2.2.2
Binary link recall	DR, R_{bin}, R_{link}	Proportion of story (passage) pair-wise links found in ground truth	4.3, 5.4, 6.2.2.2
Binary link F-value	DF, F_{bin}	Harmonic mean of *DP* and *DR*	4.4, 5.4
Link type precision	P_{type}	Proportion of correct link type found in system output	5.4
Link type recall	R_{type}	Proportion of links found with correct type in ground truth	5.4
Link type F-value	F_{type}	Harmonic mean of P_{type} and R_{type}	5.4
Joint F-value	JF	Harmonic mean of *CF* and *DF*	4.4
Relation matrix similarity	$M(R,RT)$	Similarity of relation matrices *R* (system) and *RT* (truth), with weight adjustment	4.4
Incident concentration	*Concentration*	Degree of concentration for passages in an incident (ground truth)	6.2.2.1
Cluster purity	*Purity*	Degree of purity for passages in a cluster (system output)	6.2.2.1
Link direction error	Err_{link}	Number of links in the opposite direction	6.2.2.2

Mean cluster quality	$Mean_{cluster}$	Harmonic mean of *concentration* and *purity*	6.2.2.3
Mean link quality	$Mean_{link}$	Combination of P_{link}, R_{link} and Err_{link}	6.2.2.3
Cluster-link mean	$Mean_{all}$	Harmonic mean of $Mean_{cluster}$ and $Mean_{link}$	6.2.2.3
Distance matrix similarity	*Sim(DT, DS)*	Similarity of distance matrices *DT* (truth) and *DS* (system), with distance relation adjustment	6.2.3

Table 20: List of All Evaluation Measures

BIBLIOGRAPHY

[ACE 2005] NIST 2005 Automatic Content Extraction Evaluation Official Results. http://www.nist.gov/speech/tests/ace/ace05/doc/ace05eval_official_results_20060110.htm (valid on 04/28/2008).

[ACE 2008] Automatic Content Extraction 2008 Evaluation Plan (ACE08). http://www.nist.gov/speech/tests/ace/2008/doc/ace08-evalplan.v1.2.pdf (valid on 04/28/2008).

[Allan et al 1998] J. Allan, J. Carbonell, G. Doddington, J. Yamron and Y. Yang. "Topic Detection and Tracking Pilot Study: Final Report." Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp 194-218, 1998.

[Allan et al 2001] J. Allan, R. Gupta and V. Khandelwal. "Temporal summaries of new topics." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 10-18, 2001.

[Allan 2002a] J. Allan. Topic Detection and Tracking: event-based information organization. Kluwer Academic Publishers, 2002.

[Allan 2002b] J. Allan. "Introduction to Topic Detection and Tracking." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 1-16, 2002.

[Allan et al 2002] J. Allan, V. Lavrenko and R. Swan. "Explorations within Topic Tracking and Detection." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 197-224, 2002.

[Allan et al 2003a] J. Allan, A. Feng and A. Bolivar. "Flexible Intrinsic Evaluation of Hierarchical Clustering for TDT." Proceedings of the twelfth international conference on Information and knowledge management, pp. 263-270, 2003.

[Allan et al 2003b] J. Allan, C. Wade and A. Bolivar. "Retrieval and Novelty Detection at the Sentence Level." Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 314-321, 2003.

[Bauer and Kohavi 1999] E. Bauer and R. Kohavi. "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants." Machine Learning, vol. 36(1-2), pp. 105-139, 1999.

[Beeferman et al 1997] D. Beeferman, A. Berger and J. Lafferty. "Text Segmentation using Exponential Models." Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pp. 35-46, 1997.

[Bekkerman et al 2005] R, Bekkerman, R. El-Yaniv and A. McCallum. “Multi-Way Distributional Clustering via Pairwise Interactions.” Proceedings of the 22nd international conference on Machine learning, pp. 41-48, 2005.

[Belkin and Croft 1992] N. J. Belkin and W. B. Croft. “Information Filtering and Information Retrieval: Two Sides of the Same Coin?” Communications of the ACM, vol. 35(12), pp. 29-38, 1992.

[Berger and Lafferty 1999] A. Berger and J. Lafferty. “Information retrieval as statistical translation.” Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 222-229, 1999.

[Blum and Mitchell 1998] A. Blum and T. Mitchell. “Combining Labeled and Unlabeled Data with Co-Training.” Proceedings of the Workshop on Computational Learning Theory, pp. 92-100, 1998.

[Brants 2000] T. Brants. “TnT – a Statistical Part-of-Speech Tagger.” Proceedings of the Sixth Applied Natural Language Processing Conference, 2000.

[Brown and Yule 1983] G. Brown and G. Yule. Discourse Analysis. Cambridge University Press. 1983.

[Callan et al 1992] J. Callan, W. B. Croft and S. Harding. “The INQUERY Retrieval System.” Proceedings of the 3rd International Conference on Database and Expert Systems Application, pp. 78-83, 1992.

[Cerny 1985] V. Cerny. “A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm.” Journal of Optimization Theory and Applications, vol. 45, pp. 41-51, 1985.

[Chen and Ku 2002] H.-H. Chen and L.-W. Ku. “An NLP & IR Approach to Topic Detection.” In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 243-264, 2002.

[Cieri et al 2002] C. Cieri, S. Strassel, D. Graff, N. Martey, K. Rennert and M. Liberman. “Corpora for Topic Detection and Tracking.” In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 17-32, 2002.

[Cohen 1960] J. Cohen. “A coefficient of agreement for nominal scales.” Educational and Psychological Measurement, vol. 20, pp. 37-46, 1960.

[Connell et al 2004] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah and J. Allan. “UMass at TDT 2004.” Proceedings of TDT 2004, 2004.

[DeJong 1979] G. DeJong. "Prediction and Substantiation: A New Approach to Natural Language Processing." Cognitive Science, vol. 3(3), pp. 251-273, 1979.

[Dharanipragada et al 2002] S. Dharanipragada, M. Franz, J. S. McCarley, T. Ward and W.-J. Zhu. "Segmentation and Detection at IBM." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 135-148, 2002.

[Doddington et al 2004] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel and R. Weischedel. "The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation." Proceedings of LREC 2004, pp. 837-840, 2004.

[Eichmann and Srinivasan 2002] D. Eichmann and P. Srinivasan. "A Cluster-Based Approach to Broadcast News." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 149-174, 2002.

[El-Hamdouchi and Willet 1989] A. El-Hamdouchi and P. Willet. "Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval." The Computer Journal, vol. 32(3), pp. 220-227, 1989.

[Feng and Allan 2005] A. Feng and J. Allan. "Hierarchical Topic Detection in TDT-2004." CIIR Technical Report, UMass Amherst, 2005.

[Feng and Allan 2007] A. Feng and J. Allan. "Finding and Linking Incidents in News." Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management, pp. 821-829, 2007.

[Filatova and Hovy 2001] E. Filatova and E. Hovy. "Assigning time-stamps to event-clauses." Proceedings of the workshop on temporal and spatial information processing, vol. 13, pp. 1-8, 2001.

[Fiscus and Doddington 2002] J. Fiscus and G. Doddington. "Topic Detection and Tracking Evaluation Overview." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 17-31, 2002.

[Fiscus and Wheatley 2004] J. Fiscus and B. Wheatley. "Overview of the TDT 2004 Evaluation and Results." Topic Detection and Tracking 2004 Evaluation Workshop, NIST, Dec 2-3, 2004.

[Fleiss 1971] J. L. Fleiss. "Measuring nominal scale agreement among many raters." Psychological Bulletin, vol. 76(5), pp. 378-382, 1971.

[Fuentes and Rodríguez 2002] M. Fuentes and H. Rodríguez. "Using Cohesive Properties of Text for Automatic Summarization." JOTRI2002 – Workshop on Processing and Information Retrieval, 2002.

[Furey et al 2000] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer and D. Haussler. “Support vector machine classification and validation of cancer tissue samples using microarray expression data.” Bioinformatics, vol. 16(10), pp. 906-914, 2000.

[Gee 2005] J. P. Gee. An Introduction to Discourse Analysis: Theory and Method. Routledge, 2005.

[Grishman and Hirschman 1986] R. Grishman and L. Hirschman. “PROTEUS and PUNDIT: Research in Text Understanding.” Computational Linguistics, vol. 12(2), pp. 141-145, 1986.

[Grishman and Sundheim 1996] R. Grishman and B Sundheim. “Message Understanding Conference – 6: A Brief History.” Proceedings of the 16th International Conference on Computational Linguistics (COLING), pp. 466-471, 1996.

[Harman 2002] D. Harman. “Overview of the TREC 2002 Novelty Track.” The Eleventh Text Retrieval Conference, NIST, Nov 19-22, 2002. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 314-321, 2003.

[Hartigan 1975] J. A. Hartigan. Clustering Algorithms. John Wiley & Sons Inc., 1975.

[Hearst 1994] M. A. Hearst. “TextTiling: A Quantitative Approach to Discourse Segmentation.” Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 9-16, 1994.

[Jain and Dubes 1988] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, 1988.

[Jaynes 1957] E. T. Jaynes. “Information Theory and Statistical Mechanics.” Physics Reviews, vol. 106, pp. 620-630, 1957.

[Jeon et al 2006] J. Jeon, W. B. Croft, J. Lee and S. Park. “A Framework to Predict the Quality of Answers with Non-Textual Features.” Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval, pp. 228-235, 2006.

[Ji and Grishman 2005] H. Ji and R. Grishman. “Improving Name Tagging by Reference Resolution and Relation Detection.” Proceedings of the 43rd Annual Meeting of the ACL, pp. 411-418, 2005.

[Joachims 2002] T. Joachims. “Learning to Classify Text Using Support Vector Machines.” Dissertation, Cornell University, Kluwer Academic Publishers, 2002.

[Kilander 1995] F. Kilander. “A Brief Comparison of News Filtering Software.” Unpublished paper, http://citeseer.ist.psu.edu/38985.html (valid on 04/28/2008), 1995.

[Kirkpatrick et al 1983] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. “Optimization by Simulated Annealing.” Science, Vol. 220(4598), pp. 671-680, 1983.

[Klavans and Kan 1998] J. Klavans and M. Kan. “Role of verbs in document analysis.” Proceedings of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics, pp. 680-686, 1998.

[Konstan et al 1997] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon and J. Riedl. “GroupLens: Applying Collaborative Filtering to Usenet News.” Communications of the ACM, vol. 40(3), pp. 77-87, 1997.

[Kumaran and Allan 2004] G. Kumaran and J. Allan. “Text Classification and Named Entities for New Event Detection.” Proceedings of SIGIR '04, pp. 297-304, 2004.

[Landis and Koch 1977] J. R. Landis and G. G. Koch. “The measurement of observer agreement for categorical data.” Biometrics, vol. 33, pp. 159-174, 1977.

[Lang 1995] K. Lang. “NewsWeeder: Learning to Filter Netnews.” Proceedings of the 12th International Conference on Machine Learning, pp. 331-339, 1995.

[Langley and Simon 1995] P. Langley and H. A. Simon. “Applications of Machine Learning and Rule Induction.” Communications of the ACM, vol. 38(11), pp. 54-64, 1995.

[Lee et al 2005] C. Lee, Z. Jian and L. Huang. “A Fuzzy Ontology and Its Application to News Summarization.” IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 35(5), pp. 859-880, 2005.

[Leek et al 2002] T. Leek, R. Schwartz and S Sista. “Probabilistic Approaches to Topic Detection and Tracking.” In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 67-84, 2002.

[Levow and Oard 2002] G. Levow and D. W. Oard. “Signal Boosting for Translingual Topic Tracking.” In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 175-196, 2002.

[Li and Croft 2005] X. Li and W. B. Croft. “Novelty detection based on sentence level patterns.” Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 744-751, 2005.

[Mani and Maybury 1999] I. Mani and M. T. Maybury. Advances in Automatic Text Summarization. MIT Press, 1999.

[Manmatha et al 2002] R. Manmatha, A. Feng and J. Allan. “A Critical Examination of TDT’s Cost Function.” Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pp. 403-404, 2002.

[McKenna and Liddy 1999] M. McKenna and E. Liddy. “Multiple & single document summarization using DR-LINK.” TIPSTER Text Phase III Proceedings October 96 – October 98, pp. 215-222, 1999.

[McKeown et al 2002] K. R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, C. Sable, B. Schiffman, and S. Sigelman. “Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster.” Proceedings of the Human Language Technology Conference, 2002.

[Metzler and Croft 2004] D. Metzler and W. B. Croft. “Combining the Language Model and Inference Network Approaches to Retrieval.” Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval, 40(5), 735-750, 2004.

[Mock 1996] K. J. Mock. “Hybrid Hill-Climbing and Knowledge-Based Techniques for Intelligent News Filtering.” The 13th National Conference on Artificial Intelligence, Part 1 (of 2), pp. 48-53. 1996.

[Murdock 2006] V. Murdock. “Aspects of Sentence Retrieval.” Ph.D. Dissertation, UMass Amherst, 2006.

[Nallapati et al 2004] R. Nallapati, A. Feng, F. Peng and J. Allan. “Event Threading within News Topics.” Proceedings of CIKM 2004 conference, pp. 446-453, 2004.

[Ogilvie and Callan 2003] P. Ogilvie and J. Callan. “Combining Document Representations for Known-Item Search.” Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 143-150, 2003.

[O’Leary 1997] D. E. O’Leary. “The Internet, intranets, and the AI renaissance.” Computer, Vol. 30(1), pp. 71-78, 1997.

[Olive 2005] J. Olive. “Global Autonomous Language Exploitation (GALE).” DARPA/IPTO Proposal Information Pamphlet, 2005.

[Pazzani and Billsus 1997] M. Pazzani and D. Billsus. “Learning and Revising User Profiles: The Identification of Interesting Web Sites.” Machine Learning, vol. 27(3), pp. 313-331, 1997.

[Ponte and Croft 1997] J. M. Ponte and W. B. Croft. “Text Segmentation by Topic.” Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, pp. 113-125, 1997.

[Radev et al 2005] D. Radev, J. Otterbacher, A. Winkel and S. Blair-Goldensohn. “NewsInEssence: Summarizing Online News Topics.” Communications of the ACM, vol. 48(10), pp. 95-98, 2005.

[Raghavan et al 2004] H. Raghavan, J. Allan and A. McCallum. “An Exploration of Entity Models, Collective Classification and Relation Description.” Proceedings of the Second International Workshop on Link Analysis and Group Detection, pp. 1-10, 2004.

[Robertson et al 1998] S. E. Robertson, S. Walker, M. Honcock-Beaulieu, A. Gull and M. Lau. “Okapi in TREC-7: Automatic ad hoc, filtering, VLC and interactive track.” The Seventh Text REtrieval Conference (TREC-7), NIST, 1998.

[Salton et al 1996] G. Salton, A. Singhal, C. Buckley and M. Mitra. “Automatic Text Decomposition Using Text Segments and Text Themes.” Proceedings of the seventh ACM conference on Hypertext, pp. 53-65, 1996.

[Schank and Abelson 1977] R. C. Schank and R. P. Abelson. Scripts, Plans, Goals, and Understanding: an Inquiry into Human Knowledge Structure. Lawrence Erlbaum Associates, 1977.

[Schapire and Singer 2000] R. E. Schapire and Y. Singer. “BoosTexter: A Boosting-based System for Text Categorization.” Machine Learning, vol. 39(2/3), pp. 135-168, 2000.

[Schiffrin et al 2001] D. Schiffrin, D. Tannen and H. E. Hamilton. Handbook of Discourse Analysis. Blackwell, 2001.

[Schultz and Liberman 2002] J. M. Schultz and M. Y. Liberman. “Towards a ‘Universal Dictionary’ for Multi-Language IR Applications.” In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 225-242, 2002.

[Soderland et al 1995] S. Soderland, D. Fisher, J. Aseltine and W. Lehnert. "CRYSTAL: Inducing a Conceptual Dictionary." Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1314-1319, 1995.

[Soderland and Lehnert 1994] S. Soderland and W. Lehnert. “Corpus-Driven Knowledge Acquisition for Discourse Analysis.” Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), vol. 1, pp. 827-832, 1994.

[Soboroff 2004] I. Soboroff. “Overview of the TREC 2004 Novelty Track.” The Thirteenth Text Retrieval Conference, NIST Special Publication: SP 500-261, Nov 16-19, 2004.

[Sparck Jones 1972] K. Sparck Jones. “ A statistical interpretation of term specificity and its application in retrieval.” Journal of Documentation, vol. 28(1), pp. 11-21. 1972.

[Steinbach et al 2000] M. Steinbach, G. Karypis and V. Kumar. “A Comparison of Document Clustering Techniques.” KDD Workshop on Text Mining, 2000.

[Steinberger et al 2005] R. Steinberger, B. Pouliquen and C. Ignat. “Navigating multilingual news collections using automatically extracted information.” Proceedings of the 27th International Conference on Information Technology Interfaces, pp. 25-32, 2005.

[Strassel and Glenn 2004] S. Strassel and M. Glenn. “Creating the TDT5 Corpus and 2004 Evaluation Topics at LDC.” Topic Detection and Tracking 2004 Evaluation Workshop, NIST, Dec 2-3, 2004.

[TDT2004] NIST. “The 2004 Topic Detection and Tracking (TDT2004) Task Definition and Evaluation Plan.” Topic Detection and Tracking 2004 Evaluation Workshop, NIST, Dec 2-3, 2004.

[Tong and Chang 2001] S. Tong and E. Chang. “Support vector machine active learning for image retrieval.” Proceedings of the ninth ACM international conference on Multimedia, pp. 107-118, 2001.

[Trieschnigg and Kraaij 2005] D. Trieschnigg and W. Kraaij. “Scalable hierarchical topic detection: exploring a sample based approach.” Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 655 – 656, 2005.

[van Dijk 1980] T. A. van Dijk. Macrostructures. Lawrence Erlbaum Associates, 1980.

[van Dijk 1983] T. A. van Dijk. “Discourse Analysis: Its Development and Application to the Structure of News.” The Journal of Communication, 33(2), pp. 20-43, 1983.

[van Dijk 1988] T. A. van Dijk. News as Discourse. Lawrence Erlbaum Associates, 1988.

[van Rijsbergen 1979] C. J. van Rijsbergen. Information Retrieval. Butterworths, 1979.

[Vapnik 1995] V. N. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.

[Wong 2002] L. Wong. “ANSES - Automatic News Summarization and Extraction System.” http://km.doc.ic.ac.uk/pr-l.wong-2002/index.html (valid on 04/28/2008).

[Willett 1988] P. Willett. "Recent trends in hierarchic document clustering: a critical review." Information Processing and Management, vol. 24(5), pp. 577-597, 1988.

[Xu and Croft 1996] J. Xu and W. B. Croft. "Query Expansion Using Local and Global Document Analysis." Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4-11, 1996.

[Yamron et al 2002] J. P. Yamron, L. Gillick, P. van Mulbregt and S. Knecht. "Statistical Models of Topical Content." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 115-134, 2002.

[Yang et al 1999] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. T. Archibald and X. Liu. "Learning Approaches for Detection and Tracking News Events." IEEE Intelligent Systems Special Issue on Applications of Intelligent Information Retrieval, vol. 14(4), pp. 32-43, 1999.

[Yang et al 2002] Y. Yang, J. Carbonell, R. Brown, J. Lafferty, T. Pierce and T. Ault. "Multi-strategy Learning for TDT." In Topic Detection and Tracking: event-based information organization, Kluwer Academic Publishers, pp. 85-114, 2002.

[Zemel and Pitassi 2000] R. S. Zemel and T. Pitassi. "A Gradient-Based Boosting Algorithm for Regression Problems." NIPS, pp. 696-702, 2000.