Incident Threading for News Passages

Ao Feng
Amazon.com
705 5th Ave S.
Seattle, WA 98104, USA
aofeng@amazon.com

James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
allan@cs.umass.edu

ABSTRACT

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. We explore *incident threading* as a possible solution to this problem. 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.

Earlier attempts at this problem have assumed that a news story covers a single topic. We move beyond that limitation to introduce *passage threading*, which processes news at the passage level. First we develop a new testbed for this research and extend the evaluation methods to address new granularity issues. Then a three-stage algorithm is described that identifies on-subject passages, groups them into incidents, and establishes links between related incidents. Finally, we observe significant improvement over earlier work when we optimize the harmonic mean of the appropriate evaluation measures. The resulting performance exceeds the level that a calibration study shows is necessary to support a reading comprehension task.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering; H.3.4 [**Systems and Software**]: Information Networks

General Terms

Algorithms, Design, Experimentation, Human Factors, Languages, Measurement, Performance

Keywords

Information overload, Automatic news organization, Incident threading, Passage threading

1. INTRODUCTION

Associated with the fast development of modern technologies, the amount of accessible information is increasing in an exponential manner [15]. Every day there is a large amount of new information available to us, and a major part is news. News comes from many different sources, including traditional media such as newspaper, radio or TV, and modern sources like the Web. Without proper arrangement of the information, one can easily become lost because of its vast size. This phenomenon is called *information overload*.

For an information acquisition task, there are tools available to help the web users: search engines provide general knowledge related to a query; question answering systems help a user find direct answers to his/her question; online forums and mail lists offer additional community-based support through human-to-human interaction. Nevertheless, news remains an area that has not been fully explored. Many websites publish a large amount of news, and some provide categorization information and/or search functions. Unfortunately, the problem of serving *interesting* news to a user remains unresolved.

It is infeasible for a user to sort through all available news 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 interesting information with the smallest cost, we desire a system that automatically processes news and converts it into a more user-efficient format.

Different people have their own ways of comprehending news information, but there are some common rules that most would follow. 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/her 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 military conflict in Kosovo. A good system should group news according to the main topic discussed.
- People remember interesting information for a long time, and care about new messages rather than repetitions, even if the repeated information is described in a different vocabulary. 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. Hypothetically, they would compare new information to memory and associate it with existing information that is correlated. It would be preferable if the system takes similar actions to link related (but not duplicate) events, because people are very likely to be interested in both (or neither).

Figure 1 shows summaries of four news reports from CNN (the text below a box is the document identifier). As we can see, three

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11...$10.00.

of them are from the same news topic ("Pope visits Cuba") and the last one is about the Monica Lewinsky scandal case. An ideal news organization, as shown in Figure 1, should place the three related reports together and display their contextual links, leaving the irrelevant information aside.

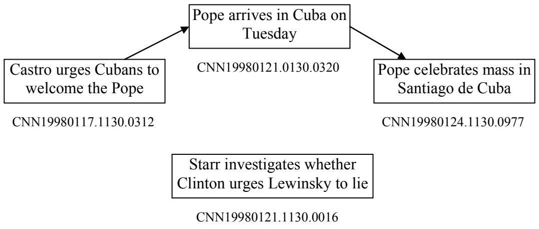
Pope arrives in Cuba on Tuesday
CNN19980121.0130.0320
Castro urges Cubans to welcome the Pope
CNN19980117.1130.0312
Pope celebrates mass in Santiago de Cuba
CNN19980124.1130.0977
Starr investigates whether Clinton urges Lewinsky to lie
CNN19980121.1130.0016

Figure 1: Sample News Organization

Our goals align with the idea of *incident threading* as proposed by Nallapati et al [14] and continued by Feng and Allan [8]. Those papers laid the foundation but were disadvantaged by making a clearly incorrect (and acknowledged) assumption that a story talks about a single incident. In this paper we extend their "story threading" work to produce "passage threading".

In the next section, we revisit the infrastructure of incident threading in the scenario of passage-based news analysis. Section 3 introduces the motivations behind incident threading, illustrates its two earlier implementations, and shows the possibility of extensions. Section 4 describes the innovative framework of *passage threading*, which conducts news analysis at the passage level. Experiments in Chapter 5 show the performance improvement of a three-stage algorithm over the earlier work. Chapter 6 summarizes contributions of the paper and proposes potential research topics for the future.

2. INCIDENT THREADING

The idea of incident threading was mainly motivated by *Topic Detection and Tracking* (TDT) [1], in which news stories are assigned to individual news topics. Each topic includes a seminal event and all directly related events. However, the discussion of how these events are organized is not the main concern of TDT. To obtain a clearer view of the news evolution, it is necessary to go beyond topics and dive deeper into their internal structure [14]. As related news is usually connected by semantic contextual information, it is desirable to establish a fact network, where each vertex represents an individual news event, and an edge shows the connection between the two events that it joins. That is the basic idea of incident threading.

2.1 Incident

We define the basic concepts before discussing the details of incident threading.

1. *News story*: Any news in text format is disseminated in units, and each of them is a news story. A story has a unique ID and a series of characters containing its content. A news story usually describes one or more real-world occurrences (events).
2. *Main characters* (WHO): The most important named entities that show who or what is involved in the description of an event.
3. *Time stamps* (WHEN): Two time features are considered for a news report. One is called publication time, which is when the news is released. The other is activity time that contains the time stamp of the described occurrence, which may be a time point or a period.
4. *Location* (WHERE): It describes where the occurrence happened, which is usually an absolute geographical position or a relative reference towards another location.
5. *Action* (WHAT): The key verb(s) in the description of the event. Sometimes it may also be a noun.

With a clear notion of these concepts, what an incident is can be naturally defined.

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

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

Basically, an incident is a real-world occurrence, which involves some named entities, and happens at a specific time and location. It can also be used to describe the union of text that contains the same (or similar) features (who, when, where, what) and describes the same episode.

2.2 Incident Network

In order to accurately model the contextual relation among incidents, we need to specify a limited vocabulary of the possible relation types. *Discourse analysis* [5, 20] provides a framework that reflects the structure of a news report, and some concepts in it can also be applied to defining the relation between two incidents.

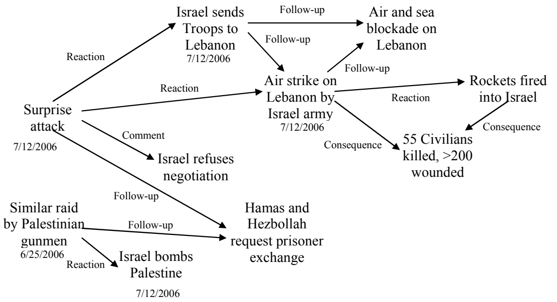
Israel sends Troops to Lebanon
7/12/2006
Follow-up
Air and sea blockade on Lebanon
Reaction
Follow-up
Follow-up
Air strike on Lebanon by Israel army
7/12/2006
Reaction
Reaction
Rockets fired into Israel
Surprise attack
7/12/2006
Comment
Israel refuses negotiation
Consequence
55 Civilians killed, >200 wounded
Consequence
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 2: Sample Incident Network

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

[1] The concept *event* is often used in Information Extraction (IE) where it has a different meaning [11], so consistent with earlier work [8] 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. The event extraction task is usually limited to certain types of events (e.g., conflicts) and its focus is on the accurate identification of their arguments. Descriptions of the same semantic content at different places are often handled separately.

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

Figure 2 shows an incident network that represents some news reports about an Israel-Lebanon conflict. The text next to each edge is the relation type of the corresponding link, and many of them are borrowed from a news schema in discourse analysis [21].

There are three main classes of connections in an incident network. Here they are described starting from the strongest type of relations.

The first class is *logical relations*. A connection of this type is established between two incidents that have logical causal relations, i.e., the occurrence of one incident directly causes the other to happen. It is represented by a directed edge in the incident network, which goes from the logical premise to the result or consequence. This class includes *Prediction*, *Comment*, *Reaction*, *Analysis*, *Background*, and *Consequence*.

The second class, here called *progressions*, requires weaker links than the previous class. In TDT, they are usually two incidents in the same topic, and one happens after another. However, one incident may not *necessarily* lead to the occurrence of the other (the link may not be causal). The only relation type in this class is named *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*. From the usual perspective, two incidents with a link in this class do not have any direct relation, except that they mention something in common. The overlapping factor may be the same main character, the same geographical location, the same type of occurrence, etc. As there is usually no priority defined by the common feature, links in this class are represented by undirected edges.

The first two classes are strong relations, because they often connect directly-related incidents to form a topic, as defined in TDT. Links in the last class usually go between topics, but they are as valuable since they connect different topics with these common factors to form a "global" incident network. This feature is especially useful to lead the user to a new topic that cannot be found otherwise.

3. PREVIOUS WORK

As we have mentioned, the idea of incident threading is motivated mostly by TDT and discourse analysis. *TDT* monitors a news stream and places the stories 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 greatly limits its application to large corpora. In this section, we also show some key decisions in the implementation of incident threading. Depending on the choices made, there can be various systems based on the incident threading framework. We briefly introduce two earlier implementations here: *story threading* and *relation-oriented story threading*. Our passage-based model will be described in the next section.

3.1 TDT

TDT 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." [2] As it involves subjective understanding of news, which may differ by human being, great difficulty is expected when the process is replicated in every detail. Several assumptions are made in TDT to reduce the complexity in its implementation.

- Topics do not overlap.
- Topics are independent.
- The internal structure of a topic does not affect evaluation results.

Starting from the pilot study in 1997 [3], there were a total of eight evaluations up to TDT-2004 [9]. The concept *topic* is empirically defined with detailed instructions, and reasonable accuracy has been observed when building topics from a continuous news stream. However, the TDT framework does not provide a clear view how a news topic is formed, plus the non-overlapping and independence assumptions of topics are often challenged.

3.2 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. However, the description of a relation is subjective. A limited vocabulary of connection types and a detailed description (ideally a definition) of each are necessary to avoid the possible confusion.

In the Information Retrieval (IR) community, we are unaware of any previous attempt before incident threading, but discourse analysis in journalism deals with similar problems [5, 20, 21]. 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.

3.3 Implementations of Incident Threading

In an incident threading system, there are two important decisions to make. The first is the selection of the basic text unit. A news story usually contains a lot of semantic information, which makes it easier to understand, but in many cases a story mentions multiple real-world occurrences. In contrast, a passage is shorter and often requires contextual information to comprehend its content entirely, but has better semantic cohesion. The second choice is on the contextual links. It is relatively simpler to determine if a relation exists between two incidents, but marking their link type may be a highly subjective task. We can either go with binary links, which are easier to annotate and implement, or require the relation type to be explicitly marked for each link.

With different answers to the two questions above, there can be four combinations in the system implementation. *Story threading* [14] selects a news story as the basic semantic unit and ignores link types. When type information is considered, it becomes *relation-oriented story threading* [8]. *Passage threading* (Section 4) analyzes news at a smaller granularity (passages instead of stories), and limits the range of news to a specific subject (violent actions in the experiment). Under that scenario, the vocabulary of relations is limited, so we choose to ignore the link types for now. We have also tried passage-level incident analysis of general news for richer relations. Unfortunately, the poor inter-annotator agreement suggests it is insufficiently understood to be tackled.

3.3.1 Story Threading

As the earliest attempt to organize news at the incident level, *story threading* [14] tries to capture the news incidents within a TDT topic and the organization among them. Incidents in the same topic are shown in a Directed Acyclic Graph (DAG). 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. Figure 3 displays the ideal incident model in this framework for the data in Figure 2.

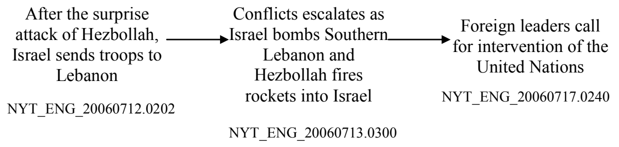
After the surprise attack of Hezbollah, Israel sends troops to Lebanon
NYT_ENG_20060712.0202
Conflicts escalates as Israel bombs Southern Lebanon and Hezbollah fires rockets into Israel
NYT_ENG_20060713.0300
Foreign leaders call for intervention of the United Nations
NYT_ENG_20060717.0240

Figure 3: Incident Model in Story Threading

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 high similarity are linked by an edge. The edge shows a "preceding" relation, as it goes from the earlier incident to the later one.

In the experiments [14], moderate accuracy can be achieved in story threading with simple algorithms and easy-to-extract features, but the simplifications in this model leave ample space for further development.

3.3.2 Relation-Oriented Story Threading

Understanding the contextual information in news reports seems straightforward to a normal person, but 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.

Fortunately, 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 generally fixed. Schank and Abelson [18] find similar phenomena in the understanding of human knowledge, and they create scripts for scenarios in real life (e.g., restaurant script[2]). 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.

After defining the link types, contextual information can be represented more accurately in an incident network [8]. The incident network composed of the stories in the Israeli-Lebanon conflict is shown in Figure 4.

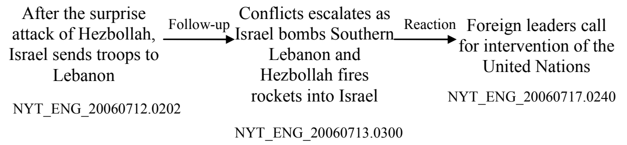
After the surprise attack of Hezbollah, Israel sends troops to Lebanon
NYT_ENG_20060712.0202
Follow-up
Conflicts escalates as Israel bombs Southern Lebanon and Hezbollah fires rockets into Israel
NYT_ENG_20060713.0300
Reaction
Foreign leaders call for intervention of the United Nations
NYT_ENG_20060717.0240

Figure 4: Incident Model in Relation-Oriented Story Threading

To establish the incident network, the same two-step process is used to create the incidents and build the links. In the second step, type-specific rules are used to assign the appropriate type label to each link. Another method is to consider the possible relations between any story pair, and expand the pair-wise competition process to a global optimization problem. For a collection of *n* stories, an *n***n* relation matrix is formed that defines a global score function, and simulated annealing is applied to find a global maximum for it.

From the experimental results [8], the revised two-step algorithm has moderate performance in the incident formation step, but creates links of low accuracy. In contrast, global optimization usually returns clusters of 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 in forming the structure of the incident network, but the high computational complexity restricts its application. The main disadvantage of the two-stage algorithm, as shown by failure analysis, is that it cannot correct clustering errors in later steps. That observation also makes clustering the performance bottleneck.

4. PASSAGE THREADING

From the beginning of TDT, a one-event-per-story assumption has been consistently applied. That assumption allows earlier research to provide useful insights in automatic news analysis. However, it is clearly not always true, despite its effectiveness in reducing the complexity of the problem. Sometimes it also introduces unsolvable problems without removing that assumption first.

For the example in Figure 2, a single news story NYT_ENG_20060712.0202 contains multiple incidents in the network:

- Surprise attack on Israeli troops.
- Israel sends troops to Lebanon.
- Hamas and Hezbollah request prisoner exchange.
- Israel refuses peace negotiation.

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

If we treat the story as a whole, the semantics relation among these incidents cannot be modeled, as it is not allowed to have a relation from one incident to itself. These links can only be appropriately formed to establish an incident network after the story is broken into smaller pieces.

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 assumed that sufficient context has been included in the story and background information is not 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[3]. 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.

4.1 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 assigned to measure each algorithm's performance.

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, we elect to use the existing text boundaries instead of applying segmentation algorithms [4, 12]. For the choice between sentences and paragraphs, we opt to treat each paragraph as an independent semantic entity and the basic unit in annotation, because it contains more information to help understand the content. With this selection, the whole annotation process and all evaluation measures treat "paragraphs" and "passages" as equivalent. There are cases where one paragraph discusses multiple incidents, but considering them makes it very difficult to achieve good inter-annotator agreement. Therefore, we make an assumption that each paragraph is an indivisible semantic unit.

We use part of the Global Autonomous Language Exploitation (GALE) [16] corpus in our data annotation, in English newswire reports only. In this collection, queries are issued to collect information on specific subjects, which are slightly different but similar to the traditional TDT topics. Some of these queries are general and collect all information related to certain topics or persons/organizations. Others focus more on special scenarios and further limit the range with query arguments. 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 from such data would be widely distributed on multiple subjects without a coherent theme, 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. Research results in such a topic can be further expanded to other categories, making it generally interesting.

In this experiment, we focus on queries that relate to "strong" or "violent" activities in news, which is one of the main topics in the GALE collection with broad interest. The selected queries are matched against the index of English newswire collections, and the 10 most similar documents are selected for annotation. The purpose of starting with queries is to provide a set of documents that are likely to be on the same topic, as well as some non-relevant documents to supply background noise. Using an actual retrieval step also makes the process more realistic: it is being applied to the output of an information retrieval system, not a set of hand-selected documents.

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.

Statistics of the annotation are displayed in Table 1. 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.

Table 1: Statistics of Annotated Corpus in Passage Threading

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)

As the annotation is subjective in the last two steps, we calculate the inter-annotator agreement only for the first step. Fleiss' Kappa [10] among 4 annotators is 0.595, and Cohen's Kappa [7] for any two annotators is between 0.548-0.664. Although there is no universal definition of a "good" Kappa value, these numbers show fair agreement among annotators, and it would be safe to claim that the problem definition is clear enough for the annotators to make rational choices. This also displays the advantage of annotating a specific topic instead of a general area, as Fleiss' Kappa in a similar annotation attempt for general incidents is only 0.193, and Cohen's Kappa ranges from 0.105 to 0.445.

[3] 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 multiple paragraphs.

4.2 Evaluation

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-generated 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 metric for system comparison. For a good estimate of the overall performance, these individual measures are combined to generate a total score.

We evaluate the clustering performance using *concentration* and *purity* scores. Suppose that an incident *I* includes *p* passages from the annotation. There are *n* clusters in the system output, and the numbers of passages in each cluster that belong to incident *I* are $p_1, p_2, ..., p_n$, respectively. These numbers should add up to *p*. Concentration for incident *I* is then defined as

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

If the passages in an incident are evenly divided into two clusters, the concentration score of the incident is approximately 0.5. The score is 0 if every cluster contains 0 or 1 passage in the incident.

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.

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

Purity is defined in the other way that measures the distribution of incidents in a cluster. If there are *q* passages in a cluster *C*, and the number of passages in it that belong to the annotated *m* incidents are $q_1, q_2, ..., q_m$, respectively (their sum may not be equal to *q*, as there can be passages that do not belong to any incident. Refer to the first step of the annotation process), the purity score for cluster *C* is

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

All purity scores of clusters with size larger than 1 are averaged to compute the overall purity.

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

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

A direct link evaluation is non-trivial as the clusters do not always match the incidents exactly. Therefore, we take an alternative approach by assuming that the links exist between passages instead of incidents or clusters. If there are *s* passages, an *s*s* matrix *M* is formed, and an element in it is

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

With the definition above, two link matrices can be easily generated, one (*MT*) comes from the ground truth, and the other (*MS*) from the system output. Then the pair-wise link precision and recall will be defined as simple matrix calculations, but they can also be expressed as "when there is a link in *MS*, chance of finding a link at the corresponding place in *MT*" and vice versa.

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

In the calculation above, a special case is ignored when the corresponding element is 1 in one matrix but -1 in the other. The proportion of arrows pointing to the wrong direction can also be calculated with these link matrices.

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

These evaluation matrices are combined into a single measure to facilitate comparison.

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

$$Mean_{all} = \frac{2 \times Mean_{cluster} \times Mean_{link}}{Mean_{cluster} + Mean_{link}} \qquad (1)$$

Inspired by the matrix comparison method in the link evaluation, more complex relations can also be encoded in a matrix. 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. The value table of a score function is shown below.

Table 2: Value Table of Score Function *f(a,b)*

b \ a	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

Then the score is added up throughout all locations in the distance matrix and normalized to limit the final value within the [0, 1] range.

$$SQ\_Sim(DT, DS) = \sqrt{\frac{\sum_{i,j} f(DT_{ij}, DS_{ij})}{\sum_{i,j} \max(f(DT_{ij}, DT_{ij}), f(DS_{ij}, DS_{ij}))}} \qquad (2)$$

If *DT* and *DS* are the same, the similarity score is 1. Depending on the percentage of elements in *DS* and *DT* that are identical, this evaluation measure can be in the range [0, 1].

4.3 Calibration Study

In preliminary experiments, performance evaluated by Equation 2 ranges from 0% to 70%, depending 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 proves to work.

The calibration study is designed in the following way. Given an existing query, a certain number (10) 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 served one version of the data, 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 (5 minutes in this experiment), the annotator skims 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 changes their performance in the evaluation.

Multiple annotators are required for the study. Each of them receives one version, either the original documents (ordered by time, statistics of the documents are in Table 1) or an incident network (as an image) at a certain performance level. Their results are checked against the standard answers, and a score is assigned to each.

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.
- The questions are approximately evenly distributed in the documents.
- The order of questions is rearranged so that it does not follow the order they appear in the documents.
- Search in the source documents is prohibited.
- Reading the questions before starting the timer is allowed and encouraged.

Table 3 shows the result of the calibration study for three queries. 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 (Equation 2). Because these "compressed" networks do not include all information in the original documents, an upper bound is listed for each of them, which means how many questions can be answered given unlimited time (Note that a higher score does not necessarily mean better coverage of the questions, which account for only a small portion of the source documents). Items in an italic font are incident networks that perform worse than the baseline (original documents) in the study, and the underlined ones are better than the baseline.

Table 3: Result of Calibration Study by Query

	Docs	Network 1	Network 2	Network 3	Network 4
1	4/10	6/7(21%)	*1/5(25%)*	5/6(28%)	6/6(32%)
2	3/10	3/4(19%)	*2/7(26%)*	5/7(30%)	5/8(37%)
3	2/10	2/3(19%)	3/5(24%)	4/6(26%)	6/6(34%)

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 perform better than the original documents in the 25-30% range of matrix similarity. Out of the 10 cells in Table 3 that have a similarity score of at least 20%, 8 of them show more correct answers than the baseline, even when the upper bound is lower (5-8 in comparison to 10).

5. Experiments

In this section, results from two systems are compared using the evaluation measures described in Section 4.2. The baseline algorithm is borrowed from *story threading* [14], with passages as the basic semantic units instead of news stories. The other method imitates the annotation process in Section 4.1, and establishes an incident network in three steps.

5.1 Baseline

The baseline algorithm starts with an agglomerative process, with each passage forming a singleton cluster. A passage is represented by a *tf·idf* vector, where *tf* is calculated with Okapi [17], and *idf* uses the Inquery normalized formula [6]. In each round, the most similar cluster pair, evaluated by average link, is merged. This process continues until all pairs have similarity below a predefined threshold. The final clusters become incidents.

Links are generated among the incidents, based on their similarity. If the average similarity between two incidents is over the threading threshold (lower than the one used in clustering), a directed arrow is formed that points from the earlier incident to the later one, where the order is determined by the time stamp of the earliest passage in each. If they have the same time stamp, the one that appears earlier in the news stream takes precedence.

The links created do not have type information, as the similarity between two incidents does not provide sufficient semantic information to assign a relation type. Similarly, the three-stage algorithm described below does not generate link types either.

5.2 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 co-reference; the last step creates links between the key incidents and all others. Likewise, we implement a three-stage algorithm to reproduce this process.

5.2.1 Passage Classification

First, a binary classifier is trained to separate "violent" passages from others. We tried SVM [22], MaxEnt [13] and BoosTexter [19] in this step using the following features:

- Number of terms in the passage
- Number of terms that appear in the main characters

- Number of terms describing locations
- Number of terms in the time stamps
- Percentage of action verbs that describe violence-related events
- Percentage of terms for all variances of "be," "do," "have" and "say"
- Percentage of terms that express certain extent of uncertainty, e.g., likely, may, can, often, sometimes, etc.
- Combinations of the three features above.
- The full text of the passage. It is available only to BoosTexter, as the other two do not accept text features.

The performance of the three classifiers in a leave-one (query)-out cross validation is shown in Table 4. The main advantage of BoosTexter is that it takes plain text features.

Table 4: Performance Comparison of Three Binary Classifiers

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}*

5.2.2 Incident Formation

The second stage runs a clustering algorithm on the "violent" passages. Although passages are shorter than news stories, snippets that belong to the same incident must have some overlap, either in terms or in semantics. The overlap may be identical terms used in both passages, a reference to the same person or organization, or a mention of the same geographical location, etc. These are the features used in the clustering process:

- Similarity of all terms
- Similarity of main characters
- Similarity of geographical locations
- Match between time stamps. As many passages are missing time stamps, we combine this feature with the term similarity by taking the product of them.

In the earlier work that utilizes multiple features [8], a weighted sum of similarities from various elements is calculated to determine the resemblance between two stories. Here a stricter requirement is enforced that matches in all attributes must be achieved for two passages to be declared similar. An agglomeration algorithm is applied to form the incidents.

5.2.3 Linking Incidents

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 time, involve the same geographical location or locations near each other, mention similar main characters, but often show poor term overlap. We still use the same features in the previous step, and one threshold is set for each. A link is created between two incidents only when all thresholds are met. Like the baseline algorithm, a link is binary and does not have a relation type.

5.3 Parameter Tuning

For both algorithms, there are some parameters that need to be adjusted based on a training set. The baseline algorithm involves only two parameters – the clustering threshold and a smaller link threshold. As the three-stage algorithm contains more features, the number of parameters is also larger.

1. The filtering threshold for BoosTexter
2. The clustering thresholds for term vectors, named entities and locations
3. The link thresholds for term vectors, named entities and locations

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 parameter tuning for both algorithms, one parameter is optimized within its range in each round, while others are fixed. This hill-climbing process continues until the performance does not improve with any change of a single parameter. We are aware of the risk that hill-climbing may return a local optimum. From our observation, the final parameters are usually in the right range, so we expect the performance to be close enough to the optimal solution.

A formal training/test division is necessary to justify the experiment results, as complex models usually have advantages in achieving better 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. From Table 1, we have 17 queries that have been fully annotated. They are similar to some extent, as these are all queries related to violent activities. On the other hand, their statistics are widely distributed, partially caused by the nature of the source documents, and also because of the difference in annotators. So here each query becomes an independent enough sub-collection. 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 then all others can be used for training.

5.4 Results

Two sets of parameter tuning are performed on the training set, where different evaluation criteria are optimized. When the harmonic mean in Equation 1 is used, the performance on the test set is shown in Table 5. Table 6 contains similar data, but the matrix comparison score in Equation 2 is optimized instead. Changes with an asterisk are significant improvements over the baseline by a one-tailed t-test. Note that smaller numbers are better for the link direction error. Clustering precision and recall [14] are also included in the tables for comparison with earlier experiments.

Table 5: Performance Comparison for Passage-Based Systems – $Mean_{all}$ Optimized

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%
$Mean_{all}$	0.0361	0.0654	+80.1%*
SQ_SIM(DT,DS)	19.10%	26.40%	+38.2%*

Table 6: Performance Comparison for Passage-Based Systems – *SQ_SIM(DT,DS)* Optimized

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%
$Mean_{all}$	0.0133	0.0110	-17.8%
SQ_SIM(DT,DS)	22.58%	25.05%	+10.9%

With different measures to optimize, the two systems show interesting performance patterns. In Table 5, 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 are assigned wrong directions by the three-stage method, and it does not receive a high score on the recall part. From our failure analysis, the reason is that many positive (violent) passages are erroneously filtered out in the first step.

When the matrix comparison measure is used to optimize the parameters (Table 6), the performance difference between the 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 achieves worse results in links, which also leads to a smaller cluster-link mean.

The appropriate 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 4.3 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 the understanding of a large collection of news reports. More experiments would help understand the correlation and difference between these evaluation measures.

6. CONCLUSIONS

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 news processing models has built-in deficiencies, this paper restates *incident threading*, which analyzes news based on the real-world occurrences discussed in a report and identifies contextual information among news incidents.

In this article we describe *passage threading* that extends earlier work and 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 challenges. The implementation starts with a fully-annotated data collection and appropriate evaluation measures for the new application. Then two algorithms are provided for a reference of the performance. The three-stage algorithm achieves significant improvement over the baseline when we tune on the cluster-link mean metric, but mixed performance is observed when the matrix comparison evaluation is used in training. Moreover, a calibration study shows that the current performance of an incident network is at least 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 paper has provided a detailed framework and sufficient support for additional developments. The current progress is encouraging, and further research in this direction is promising. This work has made contributions on both theoretical and technical aspects.

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 rely on the accurate modeling of semantic information represented in the short piece.

With the limitation of a single main subject, the possible types of relations are restricted in the current implementation. An expansion to general news would be ideal, although an annotation attempt for that case failed for lack of user agreement. Clearer instructions and extensive training should improve the inter-annotator agreement, making it possible to mark up general incidents. With a richer background, type-specific relation analysis is a foreseeable outcome, and it will certainly help the comprehension of news evolution at a higher level.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or

recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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