NEW EVENT DETECTION
USING CHRONOLOGICAL TERM RANKING

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Özgür Bağlıoğlu
May, 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Seyit Koçberber (Co-Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İlyas Çiçekli

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Çiğdem Gündüz Demir

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Kıvanç Dinçer

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of the Institute

ABSTRACT

CHRONOLOGICAL TERM RANKING BASED NEW EVENT DETECTION

Özgür Bağlıoğlu
M.S. in Computer Engineering
Supervisors:
Prof. Dr. Fazlı Can
Asst. Prof. Dr. Seyit Koçberber
May, 2009

News web pages are an important resource for news consumers since the Internet provides the most up-to-date information. However, the abundance of this information is overwhelming. In order to solve this problem, news articles should be organized in various ways. For example, new event detection (NED) and tracking studies aim to solve this problem by categorizing news stories according to events. Generally, important issues are presented at the beginning of news articles. Based on this observation, we modify the term weighting component of the Okapi similarity measure in several different ways and use them in NED. We perform numerous experiments in Turkish using the BilCol2005 test collection that contains 209,305 documents from the entire year of 2005 and involves several events in which eighty of them are annotated by humans. In this study, we developed various chronological term ranking (CTR) functions using term positions with several parameters. Our experimental results show that CTR in combination with Okapi improves the effectiveness of a baseline system with a desirable performance up to 13%. We demonstrate that NED using CTR has a robust performance in different versions of TDT collection generated by N-pass detection evaluation. The tests indicate that the improvements are statistically significant.

Keywords: chronological term ranking (CTR), first story detection (FSD), new event detection (NED), performance evaluation, TDT, Turkish News Test Collection (BilCol2005).

ÖZET

KRONOLOJİK TERİM AĞIRLIKLANDIRMASI YÖNTEMİYLE YENİ OLAY BULMA

Özgür Bağlıoğlu
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticileri:
Prof. Dr. Fazlı Can
Yrd. Doç. Dr. Seyit Koçberber
Mayıs, 2009

Son yıllarda İnternetteki hızlı gelişme, içeriğindeki bilgilerin sürekli artış göstermesi bu bilgilerin düzenlenmesi ihtiyacını ortaya çıkarmıştır. Ayrıca Web ortamındaki haber kaynaklarının sayısında ve bu kaynaklar tarafından yayımlanan haberlerde aşırı artış gözlenmektedir. Bu artış sonrasında bu haberlerin düzenlenmesi içerisinden yeni olayların bulunması, yeni haberlerin izleyenlerinin tespiti önemli problem haline gelmiştir. Yeni olay bulma (YOB) ve izleme haber akışlarını takip ederek, bu sorunu çözmeyi amaçlamaktadır. Haberlerde genel olarak önemli konular haberin başlarında verilmektedir. Bu gözlemden hareketle araştırmamızda YOB deneylerimizde en iyi sonucu veren Okapi benzerlik formülünün terim ağırlıklandırması fonksiyonunu değiştirerek, kelimelerin haber içindeki sırasını bu fonksiyona uyarlayarak bunu YOB sisteminde kullandık. Bu amaçla, Türkçe için hazırlanmış olan BilCol2005 derlemiyle birçok deney gerçekleştirdik. BilCol2005 deney derlemi TDT çalışmalarından esinlenerek hazırlanmıştır. Derlem 209,305 dokümandan ve seksen tanesi insanlar tarafından etiketlenmiş olaylardan oluşmaktadır. Bu çalışmada çeşitli kronolojik terim ağırlıklandırması (KTA) fonksiyonlarının, başarımı %13 kadar arttırdığı gözlenmiştir. Ayrıca KTA kullanarak yapılan YOB sisteminin BilCol2005'ten N-geçişli bulma yöntemiyle elde edilen farklı deney derlemlerinde de başarılı sonuçlar verdiği gözlenmiştir. Yapılan test sonuçlarında iyileştirmeler istatistiksek olarak kayda değer olduğu gözlenmiştir.

Anahtar Sözcükler: Türkçe haberler deney derlemi, haber portalı, kronolojik terim ağırlıklandırması, (KTA), performans değerlendirmesi, TDT, yeni olay bulma (YOB).

Acknowledgements

I am deeply grateful to my supervisor Prof. Dr. Fazlı Can, who has guided me with his invaluable suggestions and criticisms, and encouraged me a lot in my academic life. It was a great pleasure for me to have a chance of working with him. I am also grateful to my co-advisor Asst. Prof. Dr. Seyit Koçberber for his invaluable comments and contributions. I would like to address my special thanks to Asst. Prof Dr. İlyas Çiçekli, Asst. Prof. Dr. Çiğdem Gündüz Demir and Dr. Kıvanç Dinçer, for their valuable comments and offerings.

Also, I am very glad that I have been a member of Bilkent Information Retrieval Group. I would like to thank my friends, Süleyman Kardaş, H. Çağdaş Öcalan, and Erkan Uyar and for their collaborations in TÜBİTAK project "New Event Detection, Tracking and Retrospective Clustering in Web Sources" under grant number 106E014. My work was directly used in this project like those of the other members of the Bilkent Information Retrieval Group.

I am grateful to Bilkent University for providing me founding scholarship for my MS study. I would also like to address my thanks to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for its scholarship during my MS period.

Above all, I am deeply thankful to my parents and sister, who supported me in each and every day. Without their everlasting love and encouragement, this thesis would have never been completed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The computer revolution has evolved a society that feeds on information. Also the fast evolution and spread of Internet has accelerated this process. This causes lots of raw information with unorganized structure. This provides a huge amount of information available; however, we do not have knowledge to access them properly. Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of information should provide the user with easy access to the information, the user needs. This problem is referred as information need [BAE1999]. The huge amount of information on the Web causes the problem called "Information Overload." It refers to an excess amount of information being provided, making processing and absorbing tasks very difficult for the individual because sometimes we cannot see the validity behind the information. Information need and information overload may seem conflicting words but they complement each other. Information Retrieval aims to solve this puzzle by making access of necessary information easier.

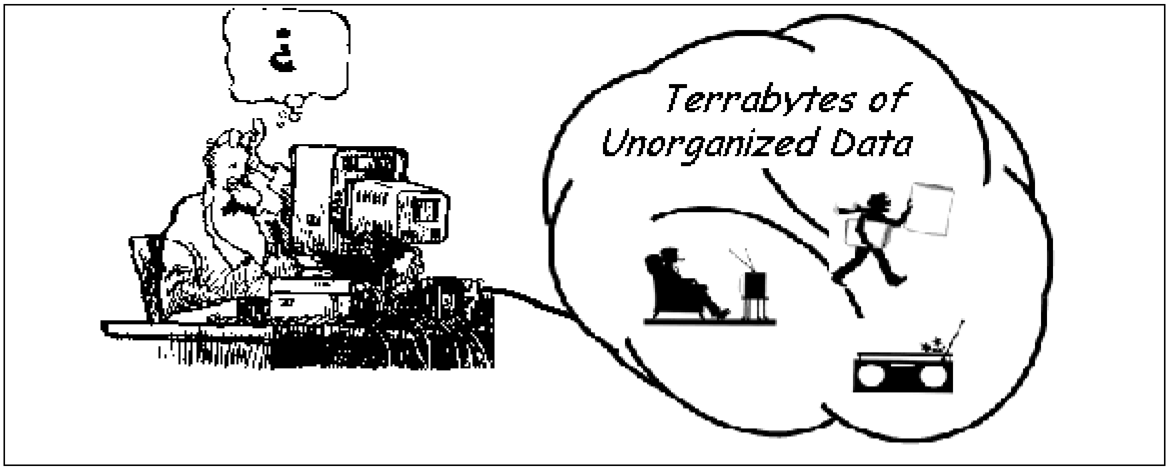


Figure 1.1: Glut of information does not guarantee more happiness.

With the advent of computer technology, it became possible to store large amounts of information; and finding useful information from collections becomes a necessity. The advances in technology indicate that the most useful information will be available in digital form within a decade. The entire corpus of published printed material produced in a year, including books, newspapers, and periodicals occupies between 50TB–200TB, depending on the compression technology [VAR2005]. With this huge information space, the processing of information and presenting in an easy way becomes very important. Information retrieval has become popular with this necessity [SIN2001]. In the past 30 years, the IR field has grown faster than expected. It goes beyond the primary goals some of which are depicted as indexing text, searching for useful documents in a collection. Nowadays, it includes modeling, document classification and categorization, data filtering, visualization, system architecture, and many other subcategories [BAE1999]. Despite its maturity, the access of relevant information is not easy. Nowadays, while accessing relevant information we also gather lots of unnecessary items. For naive users, this problem becomes more difficult. These issues have attracted the attention of the IR society, and researchers start to investigate new techniques to solve information overload problem on the Web.

One of the problems of information overload occurs in news portals which presents news articles gathered from a wide range of resources. Such portals provide lots of news with increasing amount even in small time intervals. So, the organization and presentation of information are important for usability. Although, information retrieval systems provide solutions for querying information, the news consumers should know what to query for. This can be achieved by emphasizing the presentation of relevant news. The research topics in this area include news filtering, novelty detection, news clustering, duplicate news elimination, and news categorization. In this thesis, we study the new event detection problem within the context of news portals.

Event detection is the process of discovering new events in a stream of texts. It is used in different systems such as applications for finding new trends in the stock market, detecting new problems in customer complaints, discovering stock market shifts, and detecting terrorist activities using open sources [AMI2007]. Event detection is even important to ordinary news consumers.

The goal of new event detection (NED) is to extract stories which have not previously mentioned. For instance, when a bombing event comes to news sources that have occurred recently, NED should notify users about the occurrence of this event. This problem is an instance of unsupervised binary classification where yes/no decisions are taken about the novelty of incoming event without any human interaction [PAP1999].

The initiative research for new event detection is carried by a project called Topic Detection and Tracking (TDT). According to TDT, an event is defined as something that happens at a given "place and time." It does not need to involve the participation or interaction of human actors. However according to Makkonen, this definition neglects events which either have a long lasting nature or are not tightly spatio-temporally constrained, and these events are classified as activities by Papka [MAK2004]. The definition of event is also studied by philosophers. Philosophers assert that, in a metaphysical sense, events take place when there is a conflict between physical objects [UNV1996]. Also, topic detection is a conflicting concept with event detection. The topic in TDT is defined as a seminal event or activity along all directly related events and activities. A seminal event can lead to several things at the same time and the connection between the various outcomes and the initial cause become less and less obvious as the events progress. According to Makkonen, the events that trigger lots of events may be defined as event evolution. And this seminal event is important for topic detection [MAK2003]. However, in this thesis, we consider simple events that do not trigger other events or if they trigger, all these events are dealt independently.

So from the above discussions, an event may include special elections, accidents, and natural disasters. Topic is the collection of natural disasters, elections, i.e., events [DOD1999]. Also, Papka gives another good example of event and topic: "'airplane crashes' is defined as topic however 'the crash of US Air flight 427' should be an event" [PAP1999]. From the journalist's perspective, news about an event may include i) Time, ii) Actors iii) Place, iv) How it is happened, v) Initiative Cause and vi) The impact and results [PAP1999]. The definitions and approaches described here is to model event identity. These properties give clues about solving the new event detection problem.

1.1 Motivation and Contributions

In this thesis, we explore on-line new event detection techniques in news articles. We propose a new approach that incorporates some intrinsic features of news articles for novelty scoring to existing methods to make new event detection more effective.

We firstly identify the optimum parameter sets for new event detection experiments. Then we observe that the news is written in an inverted pyramid style. It presents the most important information at the beginning of news [KEN2009]. This observation leads us to give importance to term position information. We use this as an attribute in forming document feature set and propose a new term weighting method for new event detection experiments, because NED mainly deals with news articles. To the best of our knowledge, our work is the first one that uses inverted pyramid style information in new event detection. We evaluate our approach using Turkish TDT collection (BilCol2005) prepared by Bilkent Information Retrieval Group [KAR2009, OCA2009, UYA2009].

1.2 Overview of the Thesis

In this thesis, we propose a new method for new event detection based on chronological term ranking functions within the framework of Okapi similarity measure. This thesis is organized as follows. We first review the related works in Chapter 2. The baseline new event detection process is presented in Chapter 3. In Chapter 4 improves the baseline by using chronological term ranking approach. Experimental design and results about chronological term ranking approach are presented in Chapter 5. Chapter 6 provides further experiments and discussions with chronological term ranking based NED. Chapter 7 concludes the thesis and provides promising future research directions based on the thesis work.

Chapter 2

Related Work

The new event detection and tracking in news streams is a well-known yet hot-spot research problem in the field of TDT (Topic detection and tracking). In this study, our concern is to improve the effectiveness of current new event detection techniques using position information of words in news article. By this way we propose to improve NED performance, which is referred as a hard problem in the literature of TDT by Allan et al. [ALL2000].

The most heavily studied subjects of TDT are the first story detection, i.e. new event detection (FSD), topic detection (TD), and topic tracking i.e., event tracking (TT). The most attractive and challenging task in TDT seems to be first story detection. There is a direct relationship, between the performance of first story detection and topic tracking. It is expected that a method that performs well in NED would also be an effective TT method, provided that the first stories are properly selected [ALL2000].

During NED, some of the tracking stories of old events can be incorrectly identified as the first stories of new events. Such false first stories can attract the tracking stories of already identified (true) events, i.e., cause tracking of some topics in multiple ways. So, once we improved NED systems, automatically tracking performance will develop.

In the following sections; first we give an overview of the new event detection methodologies proposed so far. Then, we also mention about the term ranking methods introduced in information retrieval studies. Lastly, we mention the structure of news article that may be valuable feature for discriminating new events in a stream of news.

2.1 New Event Detection

The aim of new event detection is to recognize when a news topic appears that had not been discussed earlier. In this thesis, new event detection (NED) and first story detection (FSD) is used interchangeably. Note that, FSD is typically approached by reducing stories to a set of features, either as a document vector [ALL1999] or a probabilistic distribution [JIN1999]. Because probabilistic distribution has not taken much attention nearly all researchers use document vectors from feature sets for NED. In the following lines all recent studies is conducted using document feature set (vector space model) approach.

Topic Detection and Tracking (TDT) is a recently founded research area that deals with the organization of information by event rather than subject. The purpose of that effort is to organize broadcast news stories by the real world events that they discuss. In this project, the news articles are gathered from various sources in parallel, and the project helped to develop an improved notion of event based topics for information organization [PAP2000].

The TDT research initiative starts in 1996 with a pilot study (DARPA, University of Massasachusetts) and continues until 2004 [TDT2008]. Originally, it is a joint effort with DARPA (US Department of Defense Advanced Research Project Agency), Dragon

Systems, Carnegie Mellon University, and the University of Massachusetts at Amherts. It was later carried out under the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program. The first TDT results is published in 1998. Several NED approaches are evaluated and studied in this research effort done by collaboration of several institutes.

University of Pennsylvania approaches the NED problem using "single-link" technique. This approach starts with each document being in one cluster and merges the clusters if they are similar enough using "nearest-neighbor" technique. The collection is processed in chronological order. If incoming document is similar to one of documents in old clusters it is labeled as old document and merged with this cluster. If similarity is below a threshold then it is labeled as new event [PAP1999].

Another research initiative, The University of Massachusetts works on a clustering approach of the news collection that returns the first document in each cluster as a result [ALL1996]. Similar documents are clustered in to the same groups of documents. Getting inspired from previous explorations of known solutions to clustering and using this approach, they detect a modified version of single-pass (making only one pass through collection) clustering algorithm for first story detection.

Carnegie Mellon University uses vector-space model to represent each document. They use general clustering techniques to represent events. A document is represented by a feature vector consisting of distinct terms with term weights being calculated using basic IR weighting approaches (tf-idf). For clustering of collection for new event detection single pass algorithm is used. These efforts are the initiative efforts for NED and these efforts formed the baseline of new event detection approaches. Also the new event detection problem has not been studied prior to the TDT research efforts [PAP1999].

The common tool applied in TDT problems is clustering. For instance, Yang et al. study new event detection problems by using hierarchical and non-hierarchical document clustering algorithms [YAN1998]. In their approach, they pay attention to temporal and content information of news articles: older documents have less influence on deciding the novelty of document. They both conduct research about retrospective – the discovery of previously unidentified event in an accumulated collection- and online new event detection –instant identification of the onset of new events from live news feeds in real time-. They use simple single pass clustering for online event detection. This algorithm processes the input documents sequentially with old documents one at a time and if the similarity between a document and an old document is below some threshold it is flagged as “new”; if incoming document’s threshold value is not below some threshold with all previous documents is flagged as “old.” Also for efficiency, they use the sliding time-window approach to decrease the number of similarity calculations (see Figure 2.1). They find that incorporating the temporal information of news articles to the process by decreasing influence of old stories improves the effectiveness of retrospective and online event detection. In their research, they conclude that on-line new event detection is somewhat more difficult than retrospective detection.

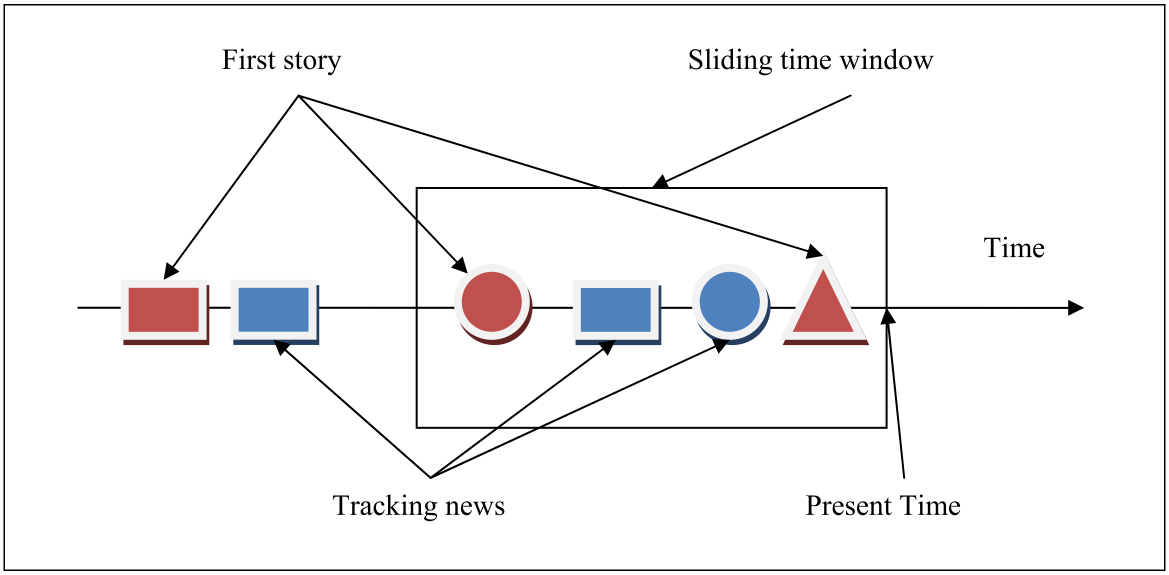


Figure 2.1: Sliding time-window approach in TDT (Different shapes represent different events).

As a part of initial TDT research initiative, in his dissertation Papka [PAP1999] also conducted experiments using a general-purpose single-pass clustering method [AND1973; RIJ1979] in various TDT-related problems such first story detection, topic tracking and clustering. He investigates the performance of the single-link, average-link, and complete-link approaches within the framework of single-pass clustering for assigning arriving stories to existing clusters. He shows that performance can be improved by using named entities and temporal information of stories. In the experiments, he uses the Inquery information retrieval system whose performance is tuned for TDT with some intuitive parameters based on experimental observations.

During NED, the newest story is compared with the earlier documents to decide if it is different (dissimilar) from them, it is treated as the first story of a new event. This decision is usually made by using a similarity threshold value that can be obtained by training. The origins of this approach can be seen in IR in single-pass document clustering [RIJ1979] or in the general cluster analysis [AND1973]. In practice, the use of such an approach is inefficient and can be unfeasible without resorting to considerable amount of hardware resources [LUO2007]. A solution to this efficiency problem is the sliding time-window concept as firstly mentioned before by Yang et al. (see Figure 2.1). In this approach, a new story is compared with only the existing members of a time-window that contains the most recent fixed number stories (or stories of a certain number of days). It is possible to use varying number of stories or days too. The time-window works like a FIFO queue. There are various possible implementations of this approach [LUO2007; PAP1999; YAN1998]. In this study, we also use the sliding time-window concept with changing number of most recent stories according to fixed time span.

Allan, Lavrenko, and Jin studied the difficulties of finding new events with the traditional single-pass clustering approach. In their work, it is shown that with certain assumptions effectiveness of one task could be predicted from the performance on the other. They show that unless there are efficient algorithms for new event detection other than single pass clustering, NED and tracking performances will not further become more effective. This is due to strong relationship between these two tasks [ALL2000].

In order to make NED system more effective, various methodologies are proposed. One of them, named composite document representation is studied by Stokes & Carty [STO2001]. They use a composite document representation that involves concept representation based on lexical chains derived from text using WordNet, syntactic representation using proper nouns, and free text representation using traditional keyword index terms to improve the online detection of new events in a stream of broadcast news. They concluded that this new representation shows performance improvements in TDT.

Another method might be to use a combination of clustering algorithms, which is studied by Yang, et al. [YAN2002]. They study a combination system called BORG (Best Overall Results Generator) by using the results of various classifiers by examining their decision error trade-off (DET) curves.

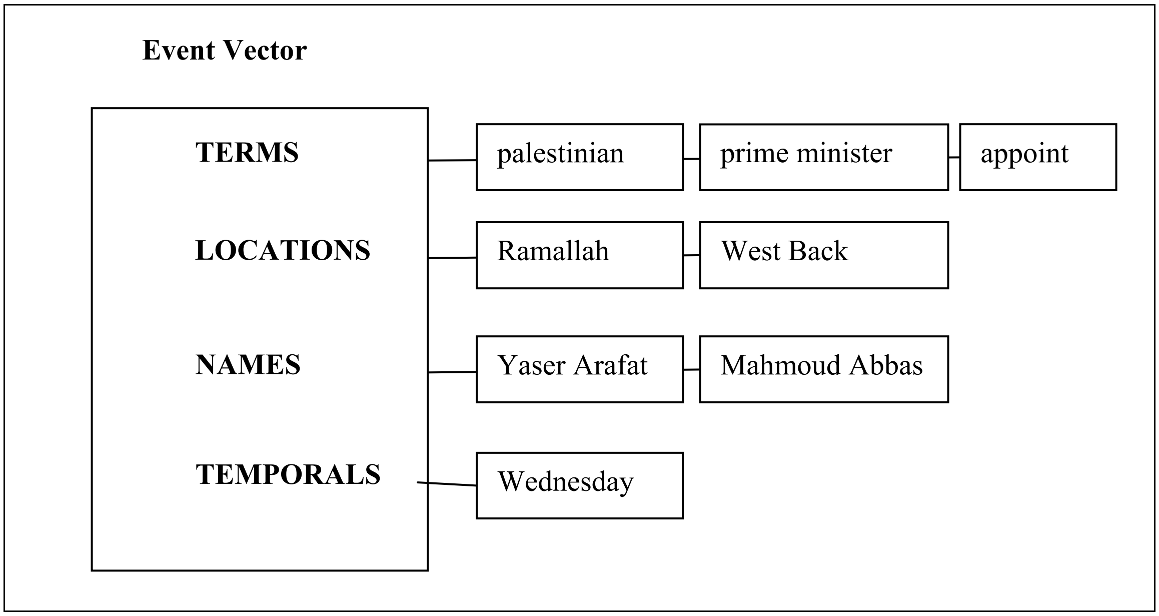


Figure 2.2: Distinct sub-vectors of document vector using simple semantics.

Yet other approach to improve NED can be seeing the effect of named entities carried out by various researchers. Kumaran and Allan [KUM2004a] examine the effects of the use of stopwords and named entities, and the combination of different document vectors (named entity vectors, non-named entity vectors) on new event detection. They present that classifying news into categories in advance improves performance. They also show that using named entities referentially is useful only in certain conditions. They also use classification techniques before NED process to improve the performance of NED. They conclude that a multi-stage NED system performs better than baseline approach. Another research about named entities is conducted by Makkonen et al. [MAK2004]. They propose a method that incorporates simple semantics into TDT by splitting the term space into groups of terms that have the meaning of the same type. They extract proper names, locations, temporal expressions, and normal terms into distinct sub-vectors of the document representation as shown in Figure 2.2. Measuring the similarity of two documents is conducted by comparing a pair of their corresponding sub-vectors at a time. They improve the performance of NED using spatial and temporal words, which are intrinsic features of news article. Lastly, the named entity approach attracts many other researchers [PAP1999; ZHA2007; KUM2005].

Brants et al. also extend baseline new event detection approaches by generating source-specific models, similarity score normalization based on document specific averages, and segmentation of stories. They use Cosine and Hellinger similarity measures. Replacing Cosine distance by Hellinger distance, source specific tf-idf model, and source specific similarity normalization provide about 18% higher performance than that of their baseline approach [BRA2003].

Efficiency issues in first story detection are studied by researchers. For instance, Luo, Tang, and Yu, conduct a research about a practical new event detection system using IBM's Stream Processing Core middleware [LUO2007]. They consider both effectiveness and efficiency of such a system in practical setting that can adapt itself according to availability of various system resources such as CPU time and memory. Luo et al. mention that their work is the first implementation of an online new event detection application in a large-scale stream processing system [AMI2007]. Efficiency issues of new event detection are also addressed in a recent work by Zhang, Zi, and Wu [ZHA2007]. They propose a new method to speed up new event detection by using an indexing tree structure. They also propose two term reweighting approaches using term type and statistical distribution distance. They conclude that their approaches significantly improve both efficiency and effectiveness.

There is little research on TDT in Turkish language. This is due to the fact that there is no standard test collection for Turkish yet. To the best of our knowledge Kurt has conducted the only TDT study for Turkish other than ours, which is conducted by the Bilkent Information Retrieval Group. He performed NED experiments using 46,530 stories belonging to the first three months of 2001 from four news resources provided by the Reuters news feed. Also, his test collection contains 15 annotated events with about 88 stories per event (min. 11, max. 358 stories) which might be statistically inadequate for effectiveness comparisons of different methodologies. The proposed method is a combination of the single-pass and k-NN clustering algorithms and uses the time-window concept. Our communication with Kurt revealed that the test collection has been misplaced and unavailable for further research [KUR2001].

2.2 News Article Format and Term Weighting Functions

In this subsection, we firstly give the general writing style of news articles which comprises our dataset for new event detection, and our application area of new event detection. Then, we give related work about term weighting functions that is generally used in information retrieval world.

2.2.1 General Structure of News Articles

The most widespread area of new event detection systems is in news portals to extract new events from articles coming from various news sources. In order to make NED process effective, the structure of news article should be examined. The news is generally written according to orientation and interests of readers. Many readers are impatient and want the events to get to the point immediately. Firstly, the reader's eye scans the headlines on a page. If the headline indicates a news story of interest, the reader looks at the first paragraph. If that also seems interesting, the reader continues. We all know that newspapers are reader oriented. So, they have to consider the scanning habit of readers about news articles to take attention for news article.

According to writing guideline of Wright, the newspaper article has all of the important information in the opening sentences. The reason is that most people do not read entire news all the way through. Also according to Bagnall, if the news article cannot get the attention in the first 8 seconds, reader won't bother with the rest [BAG1993]. Another book named "Approaches to Media Discourse" (pp 67-68) explains that the article consists of attribution (news agency, date, time, journalist's byline etc.), an abstract (lead sentence, central event of story, intro of news story, headline) and the story proper (one or more episodes) [BEL1998]. This style of writing indicates the importance of order of terms in news article properly.

To get the habits of users in mind, we now examine the writing styles of journalists. Journalists use many different kinds of frameworks for organizing stories. Journalists may tell some stories chronologically. Other stories may be read like a good suspense novel that culminates with the revelation of some dramatic piece of information at the end. Still other stories will start in the present, and then flash back to the past to fill in details important to a fuller understanding of the story. All are good approaches under particular circumstances with different categories of news articles. However the simplest and most common story structure is one called the "inverted pyramid" [KEN2009]. It forces the reporter to sum up the point of the story in a single paragraph. The inverted pyramid organizes stories not around ideas or chronologies but around facts," says journalism historian Mitchell Stephens, then continues as "It weighs and shuffles the various pieces of information, focusing with remarkable single-mindedness on their relative news value." [MIT2006].

Also it is indicated by journalists that, news writing attempts to answer all the basic questions about any particular event in the first two or three paragraphs, the Five Ws. According to journalists five Ws is a term –a formula to get the "full" answers of the story- in news writing. Five Ws (including one H) aims to answer a list of six questions, which gives important clues about events [BIL2009]:

- Who?
- What?
- Where?
- When?

- Why?
- How?

This type of structure is most common way of inverted pyramid type of writing, which refers to decreased importance of information as it progresses. The "pyramid" can also be drawn as a triangle. The triangle's broad base at the top of the figure represents the most substantial, interesting, and important information the writer means to convey. The triangle's orientation is meant to illustrate that this kind of material should head the article, while the tapered lower portion illustrates that other material should follow in the order of diminishing importance.

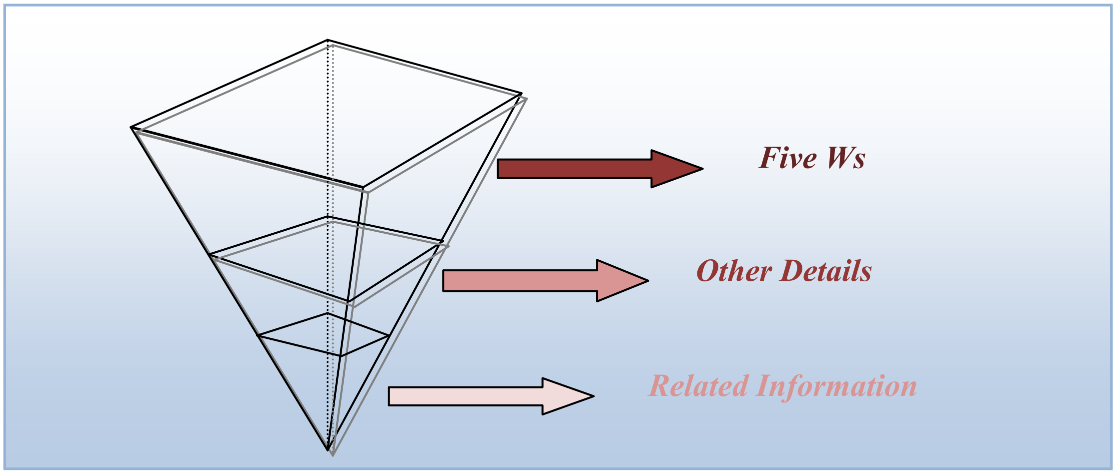


Figure 2.3: Inverted pyramid (triangle) information structure of news.

“Who”, “when”, “where”, “what”, “why” and “how” are addressed in the first paragraph. As the article continues, the less important details are presented. An even more pyramid-conscious reporter or editor would move two additional details to the first two sentences: That the shot was to the head, and that it was expected to prove fatal. The transitional sentence about the Grants suggests that less-important facts are being added to the rest of the story according to Ken Blake [KEN2009]. This type of writing also gives opportunity for editors to remove less important details of news to fit article to a fixed size. The importance goes like a triangle as in Figure 2.3.

Also, according to journalism style-guidelines the newspaper article consists of 5 parts in chronological order: headline (short attention getting statement), byline (who wrote story), lead paragraph (this is the main summary of news), explanation, additional information. According to this guideline, chronological order of a term in a document plays a key role in the identification of the document [FLY2009].

2.2.2 General Term Weighting Approaches

Term is the one of the atomic units of a document that represents the characteristics of the document. Term weighting methods assume that a term's statistical behavior within individual documents (or a collection of documents) reflects the term's ability to represent a document's content. They are also important in discrimination of a document from other documents. A term that is specific to a document can distinguish it from other documents. But some terms may appear in all collection documents. Therefore, while specific terms are of particular importance for defining a document feature set, some is the same within the whole collection.

To assess the specificity of a term within document feature set, researchers define some statistical importance values for terms. The main function of this weighting system is to enhance the retrieval performance. By using this statistical weighting approach, the document feature set becomes more discriminative for similarity calculations. Also, according to Salton "Term discrimination" suggests that the best terms for document content identification are those that are able to distinguish certain individual documents from the remainder of the collection. This definition implies that the best terms occur within a low number of documents and have high term importance within a document [SAL1988].

Term weighting functions become important when construction of the document vector. As Salton explained, terms that are frequently mentioned within the individual documents appear to be important for document feature set. Also, when high frequency terms are not concentrated in a few particular documents, but instead spread within the collection, this term has no value in document discrimination [SAL1988]. The first one implies term frequency (tf) and the second implies inverse document frequency (idf). Term weighting methods are generally based only on term statistics in complete document collection in order to appropriately weight the index terms, i.e., terms used to describe the documents. The most generally used term weighting approach in information retrieval systems are tf-idf measure, which is used to evaluate how important a word is to a document in a collection. The importance increases proportionally to the number of times a word occurs in the document but declines by the frequency of word in the corpus. It is referred as one of the most popular weighting functions in IR.

2.2.3 Relationship between Term Ranking and News Structure

The new event detection systems are generally based on information retrieval systems in term weighting of documents. They do not pay attention to the specific structure of news article. As far as we examined, there is no research in finding NED specific term weighting functions.

As we indicated before, the common application area of NED systems is on-line news event detection. The newspapers are written in inverse pyramid style shown in Figure 2.3. When taken into account, chronological of importance meaning that most important charts are in the first lines of news, details come later. So, when we give more weight to the terms occurring in the beginning of a document, we probably find a more discriminative document feature set.

2.2.4 Chronological Term Ranking

As far as we have seen in document similarity calculations, traditional tf-idf weighting model is a popular concept in information retrieval. However, not only tf-idf is an important feature in constructing document feature set. There are other metrics that are used in different systems, such as stylistic features used in authorship attribution, sentence level features in copy detection etc. The term rank in the document may also play a role in constructing document feature set. According to Troy and Zhang, the chronological term rank of a term is defined as the earliest order of term in the document [TRO2007]. The order corresponds to the order of term from beginning to end of document.

The first research about this concept is done by Troy and Zhang. Chronological rank of a term is defined as the rank of the term in the sequence of words in the original document [TRO2007]. They refer to this rank as "chronological" to emphasize its correspondence of the terms within the document from the beginning to the end. Troy and Zhang have conducted chronological term ranking (CTR) experiments using Okapi BM25. They have evaluated various combinations of CTR with Okapi BM25 in order to identify most effective CTR function.

This research is done with TREC data and topic sets consisting of Wall Street Journal (1990-1992). The research aims to improve the relevance term scoring schemas in information retrieval systems. With two different collections, MAP (mean average precision), Prec@10 (Precision after 10 documents) and reciprocal rank scores are measured using CTR functions. The scores are improved to the 5.9-26.7 percent interval with MAP, 5.8-14.9 percent interval with Prec@10, and 7.7-29.5 percent interval. But the improvements are mainly about 10-20 percent with different collections. They concluded that there is likely to be greater retrieval improvements possible using chronological ranking. They also emphasized that this work provides a good foundation for future work in the development of other approaches incorporating chronological term ranking approach.

However, information retrieval generally does not consist of news data set. So, chronological term ranking may not be suitable for all situations in retrieval systems. However, in new event detection systems the main dataset is news. The aim is to find the first occurrence of an event in a stream of news coming from different news sources. This implies that chronological term ranking is more suitable for NED systems. Also from other point, the intuition behind using chronological term rank lies in our dataset. In new event detection experiments, the dataset generally consists of news articles. In this work, we adapt chronological ranking concept to new event detection

CHAPTER 3

New Event Detection: Baseline Approach

New event detection is one of the important tasks that exist with the beginning of Topic Detection and Tracking (TDT) program conducted about more than five years [TDT2004]. NED is mainly concentrated on developing smart systems that can detect the first story on a topic of interest, where a topic is defined as “a seminal event or activity along with directly related events and activities” [ALL2002] as shown in Figure 3.1. Also according to Kumaran and Allan, a good NED system is the one that correctly identifies the news that first reports the sinking as the first story. As previously mentioned, NED has lots of practical applications such as financial markets, news analysis, and intelligence gathering where important information is usually extracted in a mass of data that grows rapidly with time [KUM2004b].

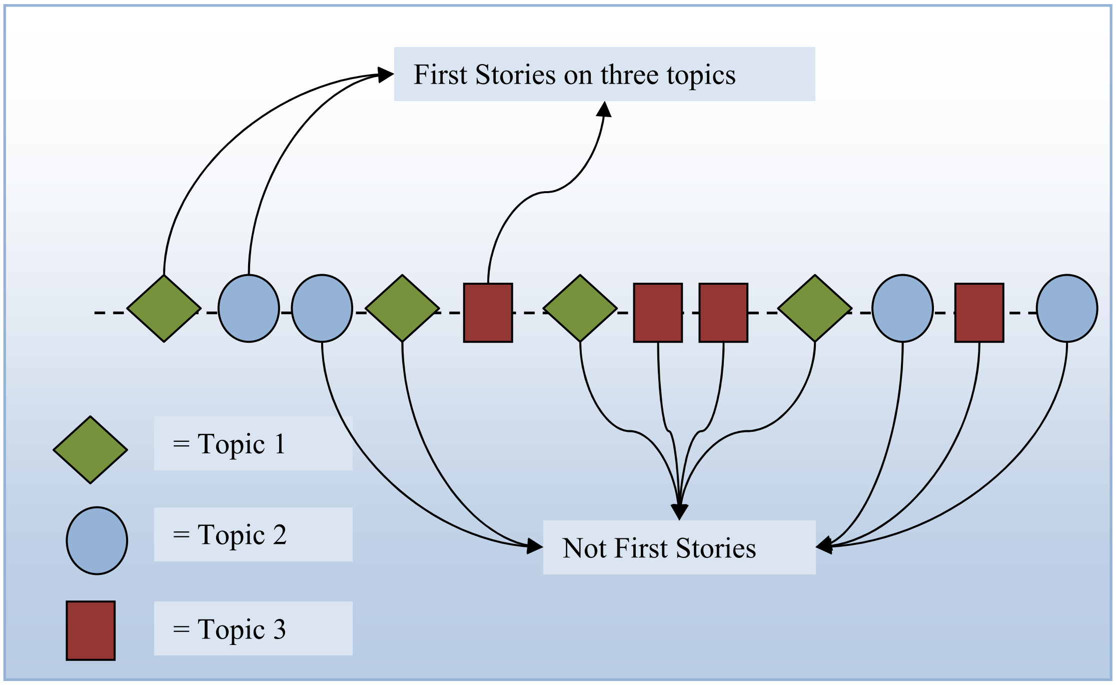


Figure 3.1: General first story detection in TDT program.

Although new event detection system is perceived as a similar task to event tracking, they have some fundamental differences. To understand event detection more clearly, we want to mention basic distinctions between two tasks. New event detection is unsupervised, that is there are no training documents or queries. Also, in NED every document must be assigned one and only one cluster in event detection systems defined as hard decisions. However, tracking is supervised, using typically 1-4 seed or training documents. Also, in tracking a document may be assigned to more than one cluster or not at all which is sometimes called soft decisions. These differences are illustrated in Figure 3.2 [FRA2001]. These differences make new event detection systems more challenging than event tracking systems.

The new event detection task is also defined as detecting, in chronological ordered stream of stories from multiple sources, the first story that discusses an event. In this task, any discussion of an event is considered as old if that topic has been already discussed in any previous story. A natural way of detecting new events is to compare the story with all old stories that have previously processed.

This task is generally done by measuring the vector based similarity between document vectors. There are a lot of ways to finding different strategies than this single pass clustering, which outperforms all others from language modeling to machine learning. Our approach mainly uses single pass clustering with vector space model [SAL1975]; we also apply some enhancements to basic NED approach performance.

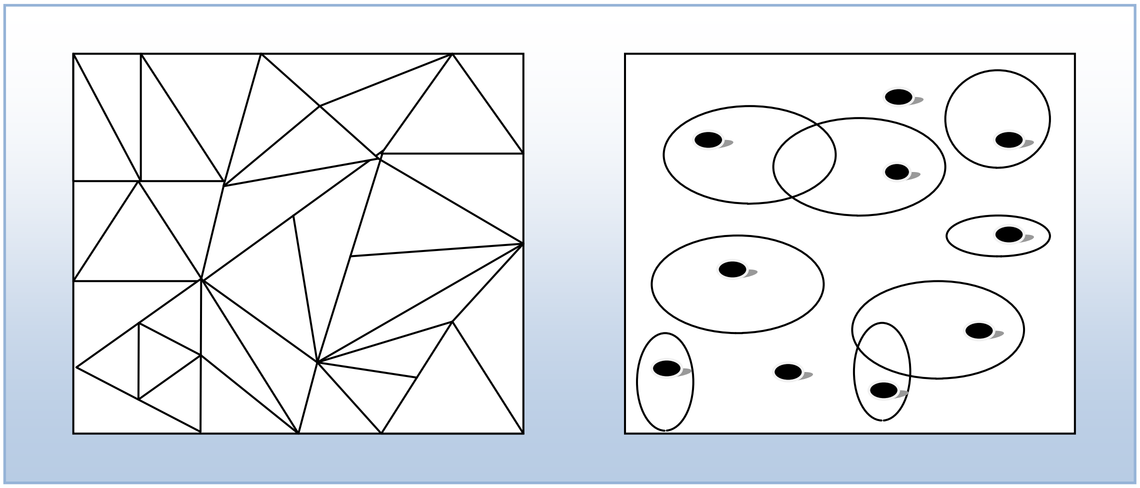
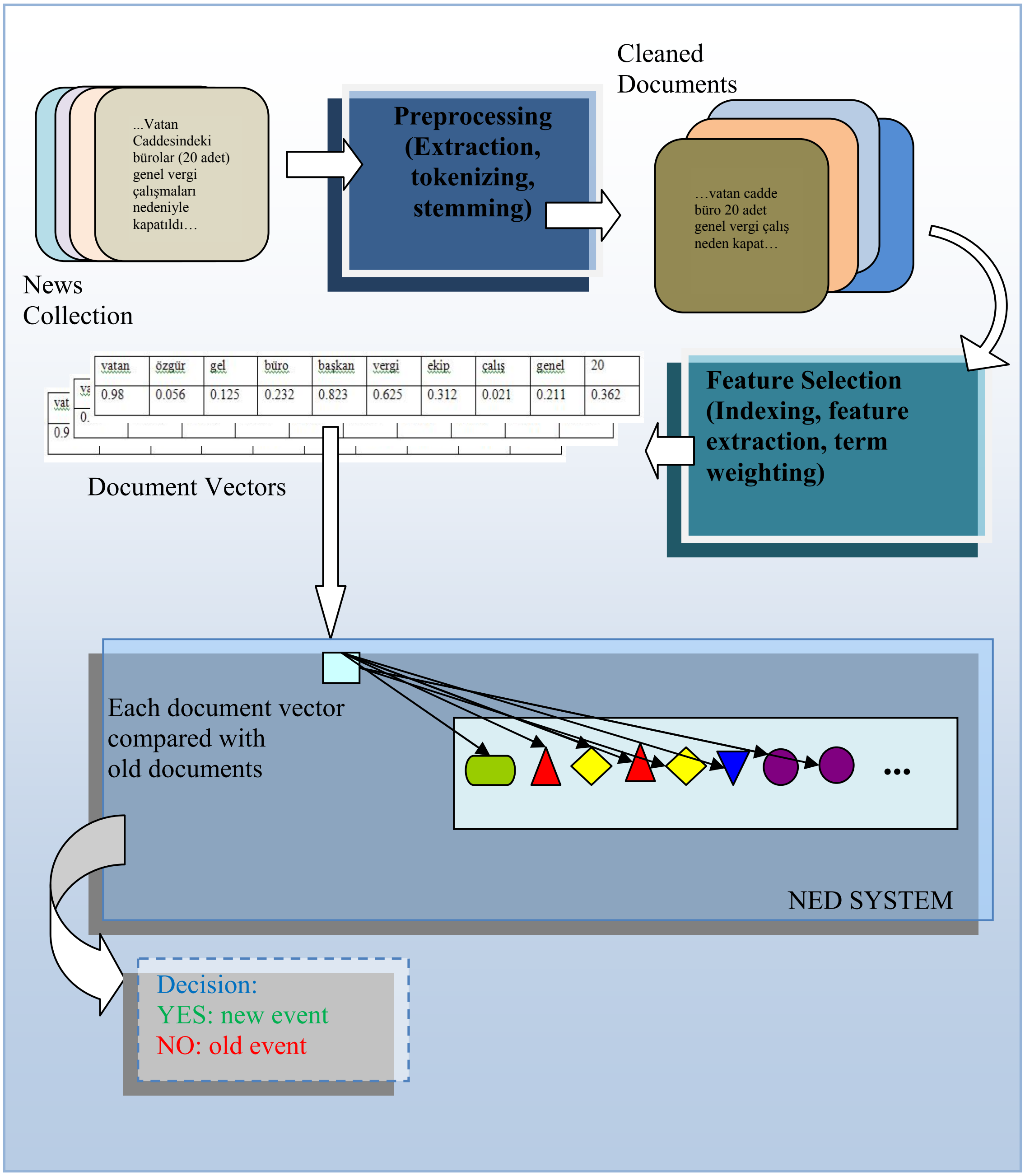


Figure 3.2: Event detection (left): unsupervised partitioning of the document space vs. Event Tracking: supervised clustering based on limited training data. Also documents in tracking may be in more than one cluster or none at all.

In the following lines, we explain the preprocessing steps of new event detection, document feature selection and similarity measures for document comparison calculations and the baseline model on detecting first stories of upcoming news as seen in Figure 3.3. We also present on-line solution to first story detection in which system indicates whether the current news article is new or old before processing subsequent story. This chapter is also important which makes a baseline for new event detection in Turkish.



vatan	özgür	gel	büro	başkan	vergi	ekip	çalış	genel	20
0.98	0.056	0.125	0.232	0.823	0.625	0.312	0.021	0.211	0.362

Figure 3.3: General system architecture of NED systems.

3.1 Preprocessing: Content Extraction

The document collection is designed as an XML (Extensible Markup Language) file structure. Sample XML file structure used in BilCol2005 is shown in Figure 3.4. The collection is already structured in chronological order in increasing of document number and time instance. When processing one document, we extract the <text> of document and process each word. While pre-processing, we eliminate some punctuation marks (question mark (?), apostrophe (‘), colon (:), semicolon (;), period (.) etc.), blanks, spaces, to get the pure word. We also change all characters of a word to lower case. We do not process the title of news. The reason behind this is that most of the news sources use very different titles with same news. After this step the document becomes a list of words with unstemmed and in the order of document ranking.

```
<DOC>
<DOCID> 0 </DOCID>
<SOURCE> Haber7 </SOURCE>
<DATE> 2005-01-01 00:00:00 </DATE>
<TITLE> Maliye gece denetiminde </TITLE>
<TEXT>
Vatan Caddesi'ndeki maliye kompleksinden saat 20.00 sıralarında ayrılan,İstanbul Defterdarlığı
Vergi Denetmenleri Bürosu Başkanı Ali Baş idaresindeki 800 kişilik denetleme ekibi,70 araçla,
gruplar halinde önceden belirlenen bölgelere dağıldı. Ekipler, Etiler, Beyoğlu ve Ortaköy başta
olmak üzere il genelindeki tüm restoran, bar ve gazino gibi eğlence yerlerinde vergi denetimi ve
belge düzenleme denetlemesi yapıyor. Kontrollerin gece boyunca süreceği ve gerçekleştirilen
denetimlerle ilgili açıklamanın daha sonra yapılacağı bildirildi. AA
</TEXT>
</DOC>
```

Figure 3.4: Sample document format from BilCol2005.

3.2 Document Feature Selection

In this subsection, we explain stopword list elimination; cleaning, tokenizing, and stemming of words; feature set construction, and document similarity calculation approaches.

3.2.1 Stopword List Elimination

Stopwords is the list of words which are generally filtered out prior to, or after, processing of natural language text. They seem to be no effect in distinguishing documents from each other. In new event detection we use three types of stopword list for evaluating the performance of each list. The first list consists of most common ten words used in Turkish. We also test the semi automatically generated stopword list of 147. The last stemming option (with 217 words) is extending these 147 words with manually found stopwords that are commonly used in Turkish language (For more information, please refer to Appendix B).

3.2.2 Stemming

After preprocessing and stopword list elimination, now comes to stemming. The purpose of stemming is to make the document representation more compact (e.g. kalemliği, kalemlik, and kalem will have one representation). In the previous work conducted by Can et al. stemming has significant effect on information retrieval systems. So, in this study we also evaluate the effects of stemming on new event detection. There are various stemming algorithms introduced by Can et al. [CAN2008a]. Similar stemming algorithms are used in this research too.

Firstly, we want to give some features of Turkish language. Turkish is an agglutinative language similar to Finnish and Hungarian. Such languages carry syntactic relations between words or concepts through discrete suffixes and have complex word structures. Turkish words are constructed using inflectional and derivation suffixes linked to a root.

In this work, we implement three stemming methods for feature extraction of document vectors. These stemmers are: no stemming (Austrich algorithm), first n (called as n-prefix) characters of word, and lemmatizer based stemmer. These approaches are shortly defined in the following lines:

- No-Stemming (NS): As the name implies, this approach uses words as they are.

- Fixed Prefix Stemming: This technique simply truncates the words and uses the first n characters of each word as its stem. When the character length of a word is less than n, it is used with no truncation. This technique is named as Fn where n defines the truncation length. In information retrieval experiments conducted by Can et al., F5 and F6 are better than other stemming options [CAN2008a]. So we also evaluated these two fixed prefix stemmers in NED experiments. As Turkish language is an agglutinative, we hope that these methods probably give satisfactory results in NED.

- Lemmatizer Based Stemming: This approach uses a morphological analyzer which explores inflected word forms and returns their dictionary forms. Lemmatizer uses more sophisticated techniques in stemming. We used lemmatizer that is developed by Kemal Oflazer [OFL1994]. There are cases which have more than one stemmer for a word. In such cases, the selection of the correct word stem (lemma) is done by using the following steps [ALT2007]. (1) Select the candidate whose length is closest to the average stem length for distinct words for Turkish; (2) If there is more than one candidate, then select the stem whose word type (POS) is the most frequent among the candidates.

We use no stemming, F5 and F6 options of fixed prefix stemming, and lemmatizer based stemming in our experiments.

3.2.3 Feature Selection

For feature selection of the document, all the words after stemming and stopword list elimination are used. We use vector space model for document representation. Each document is represented by a document vector of n document terms with the highest tf-idf score. By using the tf-idf values, we index documents by using the most representative or discriminating story terms [SAL1988]. The weight of term is calculated according to following formula:

$$w(t,\vec{d}) = (1+\log_2 tf(t,\vec{d})).\log_2(\mathrm{N}_\mathrm{t}/\mathrm{n}_\mathrm{t})$$

- $w(t,\vec{d})$: is the weight of term t in document (vector) $\vec{d}$.
- $tf(t,\vec{d})$: is the number of occurrences of term t in document d.
- $\log_2(\mathrm{N}_t/\mathrm{n}_t)$: is the IDF (inverse document frequency).
- N_t: is the number of accumulated stories so far in the collection.
- n_t: is the number of stories in the collection that contains one or more occurrence of term t up to the lastly processing document.

IDF measure is incrementally updated according to n_t and N_t at each time a new document is processed. This approach experimentally performs better than statically evaluated IDF approach, which is defined dynamically according to collection's characteristics. This approach is also used by other researchers such as [YAN1998] and [BRA2003].

For starting point of incremental IDF calculation, we use an auxiliary corpus containing the 2001-2004 news stories, about 325,000 documents of Milliyet Gazetesi that is used in IR experiments by Can et al. [CAN2008a], and update the IDF values with each incoming story. Note that this term weighting is used with most of the similarity measures. However, some similarity measures have their own tf-idf calculation formulas. These are Hellinger and Okapi, which uses similar approach to tf-idf calculation. The details of Hellinger and Okapi are given in the next subsection.

3.2.4 Similarity Calculation Method

In document similarity calculation, we conduct experiments with different similarity measures used in literature. As noted in Chapter 2, several similarity measures are proposed for NED systems. In this research for completeness and consistency of work we evaluate NED performance of various similarity functions.

TABLE 3.1: Similarity functions in NED experiments

Similarity Function	***Formula***
Cosine[1]	$\frac{\sum_{i=1}^{n} xi * yi}{\sqrt{\sum_{i=1}^{n} xi^2 * \sum_{i=1}^{n} yi^2}}$
Dice[1]	$\frac{2\sum_{i=1}^{n} xi * yi}{\sqrt{\sum_{i=1}^{n} xi^2 + \sum_{i=1}^{n} yi^2}}$
Jaccard[1]	$\frac{\sum_{i=1}^{n} xi * yi}{\sum_{i=1}^{n} xi^2 + \sum_{i=1}^{n} yi^2 - \sum_{i=1}^{n} xi * yi}$
Overlap[1]	$\frac{\sum_{i=1}^{n} xi * yi}{\min(\sum_{i=1}^{n} xi^2, \sum_{i=1}^{n} yi^2)}$

The most popular similarity measures are Cosine and Okapi used by various researchers in new event detection studies. Other similarity measures used in document comparison are Dice, Jaccard, Overlap, and Hellinger. Four of them are given in TABLE 3.1. Also there are two other similarity measures used in literature; Hellinger and Okapi functions use different types of tf-idf weighting for terms in the document. Thus, they are mentioned in the following lines.

Hellinger Similarity

In this function every term t in document d is weighted as follows at a given time:

$$weight(d,w) = \frac{1}{Z_t(d)} f(d,w).\log\frac{N_t}{df_t(w)}$$

[1] X and Y are two document vectors that are weighted according to tf-idf function.

- N_t is the total number of documents at time t.

- $Z_t(d) = \sum_w f(d,w).\log \frac{N_t}{df_t(w)}$ is the normalization value.

The similarity value between d and q documents is calculated as [BRA2003]:

$$sim(d,q) = \sum_w \sqrt{weight_t(d,w) \cdot weight_t(q,w)}$$

Okapi Similarity

In Okapi similarity, the weight of terms is evaluated differently from other similarity measures. The idea behind using Okapi in similarity calculations is that the more times t appears in a document D, and the fewer times t appears in other documents (i.e., the less popular t is in other documents), the more important t is for D [SIN2001]. Here are the details of Okapi calculation:

S is the document collection and D is a document $D \in S$. The weight of every term t in $D(w_{tf})$ is calculated as follows.

$$w_{tf} = \frac{(k_1+1)tf}{k_1\left[(1-b)+bx\frac{dl}{avdl}\right]+tf}$$

- $b = 0.75$,

- $k_1 = 1.2$,

- dl = document length,

- avdl = average document length

Inverse document frequency is calculated as follows.

$$w_{idf} = \ln \frac{N - df + .5}{df + .5}$$ [2]

- df = number of documents that includes term t,
- N total number of documents in *S*.

The similarity value between d and q documents is calculated as [LUO2007]:

$$sim(d,q) = \sum_{t \in d,q} w_{tf,d} w_{tf,q} w_{idf}$$

3.3 Baseline New Event Detection System

The purpose of first story detection is to extract stories in a stream of news that contain a discussion of new event. The new event detection algorithm sequentially processes stories in chronological order. It decides, for each incoming story, whether it is related to some existing events or discusses a new event. The decision is an instant decision, not retrospective, e.g., decision should be finished until the next incoming event process. In order to decide a new event, it is compared to all previous documents;

$$score(d_i) = \max_{d_{prev} \in S}(sim(d_i, d_{prev}))$$

- d_i, incoming document vector
- d_{prev}, previous document processed, is an element of S (news collection)

The Highest similarity of the incoming document with previous documents is identified. If the score (NED) is below some threshold, it means the incoming document is not sufficiently similar to the previous documents and it is labeled as a new event.

[2] The idf calculation is also done as incremental idf calculation similar to tf-idf approach

1. Preprocess the news document (tokenizing, stemming, etc).
2. Form the classifier representation (document vector) by feature extraction.
3. Compare the new document against existing classifiers in sliding time-window.
4. If the document does not result a high similarity score (i.e. highest similarity value is below threshold) with any existing classifier in time-window, flag the document as containing **a new event.**
5. If the document results a high similarity score (i.e. the score is above threshold.) with any existing classifier in time-window, flag the document as not containing a new event i.e. **old event.**
6. Add last story into the time window, and remove the oldest story from time-window.
7. Add the incoming document to sliding time-window and remove the oldest document from time-window.
8. Go to step 1 for processing a new document.

Figure 3.5: Baseline new event detection algorithm.

There are efficiency improvements to this basic approach. We do not compare the new incoming document with all previous documents, which may cause to some delays in decision –also most of the times it is not necessary-. So, we use a sliding time-window concept in which single pass clustering is done by using the most recent m stories. If the maximum similarity score between the incoming story and stories in the most recent m stories is below a pre-determined threshold, a flag of ‘New’ is assigned to the story. The algorithm is outlined in Figure 3.5. The confidence score for this decision is defined to be:

$$score(d_i) = \max_{d_k \in window} \left(sim \ (d_i, d_k) \right)$$

- d_i is the incoming news story,
- d_k is the kth document in the window, and k = 1, 2, 3…, m.

3.4 Evaluation Metrics used in NED

TABLE 3.2: Fundamental Evaluation Metrics in NED

Miss Rate = c / (a+c) False Alarm Rate = b / (b+d)		**Reference Annotation**	
		Target	Non-Target
System Response	YES (a Target)	Correct (a)	False Alarm (b)
	NO (Not a Target)	Missed Detection (c)	Correct (d)

Similar to the most of the systems, NED system is presented with input data and a hypothesis about the data, and the system's task is to decide whether the hypothesis about the data is true or not. The missed detection and false alarm are defined as follows.

- If hypothesis is true it is named as target, else no-target trial. According to TDT, a target story can be detected correctly as target, or it can be missed, which is named as **missed detection.**

- A non-target story can be correctly determined as non-target, or it can be falsely detected, which is depicted as **false alarm** [YAN2002].

From these definitions, if we miss a new event then miss detection increases, if we detect an old event as a new event it is called false alarm. These measures are given in TABLE 3.2.

Miss rate and false alarm are primary measures used to measure the system performance in TDT programs. There exist two techniques to represent miss rate and false alarm values. The first is decision error tradeoff curve (DET) [DOD1997], other is detection cost function (C_{det}). Detection cost function expresses performance with a single number at a particular point using actual decisions, and DET is a curve to see the tradeoff between miss rate and false alarm as shown in Figure 3.6. They are obtained by moving thresholds on detection decision confidence scores. DET curves give detailed information; however, they may be difficult to use for comparison. Thus C_{det} measure is more preferable representation of system performance than DET curve assessment.

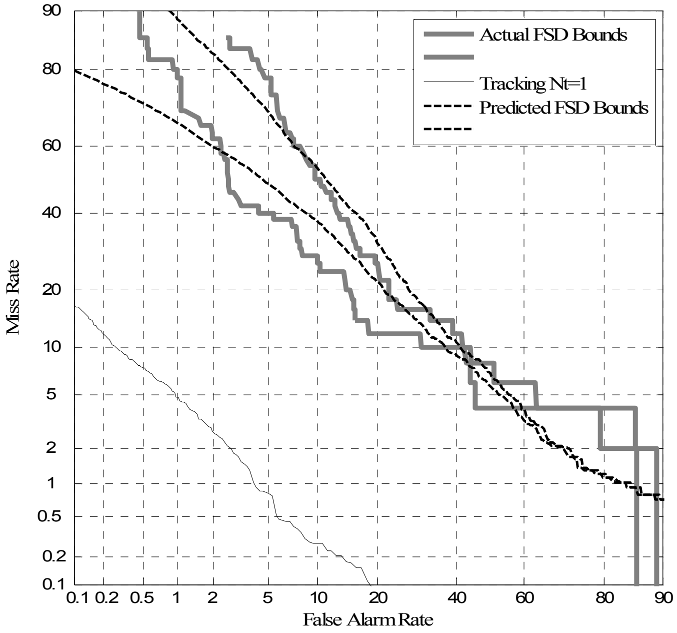


Figure 3.6: Sample DET curve representation.

C_{det} and DET use miss rate and false alarm probabilities to represent system performance. These probabilities are estimated over an evaluation data set that comprises a large number of stories and modest number of topics according to TDT [DOD1998]. For miss rate and false alarm probabilities there are two methods: story-weighted and topic weighted. According to TDT, topic weighted estimates are superior to and more suitable than story weighted estimates. So in our experiments we use topic-weighted evaluation methodology of miss and false alarm rates. The details of story-weighted and topic-weighted evaluations are given in Appendix C.

In this research we use C_{det} measure and topic weighted evaluation of error probabilities to evaluate the performances of different systems. The calculation of detection cost function is given as follows [FIS2002].

$$C_{\det} = C_{miss} \cdot P_{t\arg et} \cdot P_{miss} + C_{FA} \cdot P_{FA} \cdot (1 - P_{t\arg et})$$

- $C_{miss} = 1$, $C_{FA} = 0.1$ are the costs of a missed detection and a false alarm.
- $P_{target} = 0.02$, the a priori probability of finding a target.
- P_{miss}[3]: miss probability (rate) determined by the evaluation result.
- P_{FA}[3] : false alarm probability (rate) determined by the evaluation result.

C_{miss} C_{FA} and P_{target} are pre-specified numbers also used by TDT program which are somewhat a standard values for NED evaluations, used by all researchers in NED evaluations.

However, C_{det} has a dynamic range of values which makes difficult to interpret (i.e., good performance results in C_{det} on order of 0.001). For this reason, a normalized version of C_{det} is preferred for comparison. Detection cost function is recalculated for normalization as follows.

$$(C_{\det})_{Norm} = {C_{\det}}\Big/{Minimum\{C_{miss} \cdot P_{t\arg et}, C_{FA} \cdot (1 - P_{t\arg et})\}}$$

The values obtained by this normalized calculation of C_{det} most likely lies between 0 and 1; it can be greater than 1. The value 0 reflects the best performance that can be achieved. The value 1 corresponds to a random baseline and means that the system is doing no better than consistently guessing “no” or “yes” [FIS2002, FIS2004].

We may sum up the evaluation method as follows.

- To evaluate performance, the stories are sorted according to their scores, and a threshold sweep is performed.

[3] These values are calculated from topic-weighted method of error probabilities i.e., miss rate and false alarm probabilities.

- All stories with scores above the threshold are declared as old, while those below it are considered new.
- At each threshold value, the misses and false alarms are identified, and a cost is calculated as a linear function of their values.
- The threshold that results in the least cost is selected as the optimum one [KUM2004b].

In this research, different NED systems are compared based on their minimum cost. This minimum cost is depicted as $\min((C_{\det})_{Norm})$. In our experiments we use minimum normalized cost functions to compare different approaches. The minimum cost function is formulized as follows.

$$_{\min}(_{norm}(C_{\det})) = \min\{_{norm}(C_{\det})\}\ ,\ _{norm}(C_{\det}) \in S$$

- S, set of all normalized cost values calculated by in each threshold value using threshold sweep approach.

3.5 Experimental Dataset: Training and Test Collections

In this subsection, we describe the contents of the test collection from various perspectives, i.e., document number, number of sources, number of topics. BilCol2005 uses five different Turkish web sources while constructing its NEDT collection which publishes news from different perspectives. These sources are:

- CNN Türk with 23,644 news
- Haber 7 with 51,908 news
- Milliyet Gazetesi with 72,233 news
- TRT with 18,990 news
- Zaman Gazetesi with 42,530 news.

BilCol2005 contains a total of 209,305 documents. It contains 80 new events with their tracking stories. We divided BilCol2005 into two sets for experimental purposes. The first eight months period is served as a training set, and the last four months period is presented as a test set. For two topics used in training in BilCol2005, there exists a considerable number of news articles that also lasts during the four months period, for these two topics, their first stories (which are in fact tracking events) in the test set section are used as the first stories of the two new events. So, this makes the collection composed of a total of two sets and 82 topics all-together.

TABLE 3.3: Distributions of stories among training and test sets

Set Name	**Time Span (month.day.year)**	**No. of Topics**	**No. of Documents**	**No. of Relevant Documents**
Training	01.01.2005 - 08.31.2005	50	141,910	3,358
Test	09.01.2005 - 12.31.2005	32	67,395	2,288

TABLE 3.4: Information about distribution of stories among news sources

News Source	**# of News Stories**	**% of All Stories**	**Download Amount (MB)**	**Net Amount (MB)**	**Average No. of Words per Document**
CNNTürk	23,644	11.3	1,008.3	66.8	271
Haber 7	51,908	24.8	3,629.5	107.9	238
Milliyet Gazetesi	72,233	34.5	508.3	122.5	218
TRT	18,990	9.1	937.9	18.3	121
Zaman Gazetesi	42,530	20.3	45.3	33.7	97
All together	209,305	100.0	6,129.3	349.2	196

The details of training and test sets are shown in TABLE 3.3. Also, some statistical information about our dataset is given in TABLE 3.4. In this chapter, we used only the training data set for baseline experiments.

3.6 Baseline NED Experiments

In this subsection, we present some experiments to form the baseline NED system. The experiments are composed of stemming selection, stopword usage, similarity function selection, window size selection, and document vector length selection.

In these experiment series, we firstly determine the window size used in our further experiments. We then evaluate stemming option, a document vector size pair that performs the best with each similarity function. We have also assumed that using stopwords increases the system's performance. Lastly, we monitor the effect of various stopword lists usage in NED performance.

3.6.1 Window Size Selection

For new event detection, it would be reasonable to choose a window size that would give us a high opportunity of finding the topic cluster's two events being in the window for comparison, so that any tracking news can be compared with previous events in the same cluster and they are not falsely labeled as a new event. For this purpose, we analyze the average time distance among the stories of individual 50 topics in training set of BilCol2005. For the most (48 out of 50) of the event clusters, the average time difference among the stories of a cluster (topic) is less than 12 days. For this reason, we prefer the 12-day sliding time-window size for single pass clustering comparisons [CAN2009]. Our collection contains about 550-600 news per day and 12 days span is about 6900 news in the window. In our experiments we use a sliding time window of 7000 recent news.

3.6.2 Similarity Function Selection

The similarity functions are evaluated individually with various document vector size and stemming selections. As we depicted before, there exist six similarity functions: Cosine, Dice, Jaccard, Overlap, Hellinger, and Okapi. These functions are individually evaluated with four stemming options: First 5 (F5), First 6 (F6), lemmatizer, and no stemming (NS). Also the experiments are conducted in different document vector length selection points with 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 140, 160, 180, 200 terms and whole term list in a document. So there are 7 similarity measure, 4 stemming options, and 16 document length alternatives: This makes a total of 448 tests all. The test results are given in TABLE A.4-9. According to tests the best document vector and stemmer combination with each similarity function is given in TABLE 3.5.

TABLE 3.5: The Best document length-stemmer combinations[4]

Similarity function	**Stemmer**	**Vector Length**	**$(C_{det})_{Norm}$**
Okapi	LM	50	**0.5424**
Jaccard	LM	All Terms	0.5664
Cosine	LM	All Terms	0.5777
Overlap	F5	30	0.6573
Dice	LM	All Terms	0.5669
Hellinger	LM	All Terms	0.6207

According to results, Okapi similarity function outperforms all other functions by using 50 top terms (terms with high weight). So, we use Okapi similarity as a baseline in NED experiments in the following chapters.

We also evaluate the effect of the use of stopword list in first story detection experiments, the results indicate that using a stopword list with 217 words performs better than other options (i.e., with list with 10 words, 147 words, and no stopword list).

3.7 Chapter Summary

As a brief overview of our approach, when a document comes, firstly its feature vector is formed using most representative n terms. After defining vector space model for the document which includes stemming and stopwords removal process, the similarity is calculated between the incoming document and the most recent m documents in a sliding time-window, and if the maximum similarity is below a pre-defined threshold then it is labeled as ‘New’, otherwise it is flagged as ‘Old’.

For baseline NED, we have found that the performance of Okapi with top 50 terms as a document vector and lemmatizer as a stemmer performs the best. Also in the experiments we use a sliding time-window with 12-days span and a stopword list of 217 words.

[4] LM: Lemmatizer, F5: First Five, ALL TERMS: All document terms are used in the creation of document vectors.

Chapter 4

Chronological Term Ranking for NED

In information retrieval, vector space model is used with traditional tf-idf weighting approach [SAL1988]. The term rank in the document may also play a role in constructing document feature set, according to Troy and Zhang, the chronological term rank of a term is defined as the earliest order of term in the document [TRO2007]. The order corresponds to the order of term from the beginning to the end of the document.

The intuition behind using chronological term rank lies in the nature of our dataset. In NED applications, datasets usually consist of news articles. According to news article writing style, the importance goes like a triangle as depicted previously in Chapter 2. So, as our dataset contains news articles, we may use this valuable information for identifying new events.

4.1 Chronological Term Ranking Model

The baseline approach ignores term ranking and uses an incremental tf-idf approach with Okapi. CTR function aims to improve the relevance estimation among documents by using term position information. The word chronological emphasizes sequential occurrence of the terms within a document from the beginning to the end [TRO2007]. The intuition is that news articles tend to state the main ideas as soon as possible. Existing term frequency based (tf-idf) NED systems most likely neglect the significance of CTR.

The CTR function is modeled as follows.

Let $D = (t_1, t_2 ..., t_n)$ be a document where t_i are terms (words) ordered according to their sequence in the original document. Let tr_i*=i,* where the chronological rank *tr* of term *t* is assigned as the subscript *i* of the earliest occurrence of *t* in *D* [TRO2007].

In the following lines we propose some functional considerations to basic CTR functions to utilize the similarity calculations with our Okapi's term ranking model.

4.2 Enhancements in CTR

Our aim is to construct a more effective relevance ranking formula with compared with no-use of position information. For this purpose, we introduce a variety of term weighting functions in Okapi.

In Okapi the weight of a term t is calculated as follows.

$$w_{tf} = \frac{(k_1 + 1.0)tf}{k_1\left[(1.0 - b) + bx\frac{dl}{avdl}\right] + tf}$$

where b= 0.75, $k_1 = 1.2$, dl = document length, and avdl = average document length are standard parameters of term weighting formula.

Inverse document frequency component of Okapi is calculated as follows.

$$w_{idf} = \ln \frac{N - df + 0.5}{df + 0.5},$$

where df is the number of documents containing term t; N is the total number of documents in the collection at a given time. The similarity between document d and q is calculated as follows [LUO2007].

$$sim(d,q) = \sum_{t \in d,q} w_{tf,d} \cdot w_{tf,q} \cdot w_{idf}$$

Our approach to integrating CTR feature to the Okapi formula is similar to that of Troy and Zhang but more comprehensive [TRO2007]. We integrate the CTR component $R_{t,d}$ to the Okapi formula in two ways by using

- Additive functions

$$sim(d,q) = \sum_{t \in d,q} \left(w_{tf,d} + R_{t,d} \right) \cdot \left(w_{tf,q} + R_{t,q} \right) \cdot w_{idf}$$

- Multiplicative functions

$$sim(d,q) = \sum_{t \in d,q} \left(w_{tf,d} \cdot R_{t,d} \right) \cdot \left(w_{tf,q} \cdot R_{t,q} \right) \cdot w_{idf}$$

There are some concerns in the formulation of $R_{t,d}$.

- The term rank can be used as an inverted absolute rank: C/tr or as a percentage rank: $C \cdot \frac{tr}{dl}$, where C is a constant.
- If document length is used for term ranking, we may either use the actual document length or the maximum document length seen so far not to overemphasize the CTR values of short documents.
- We may use logarithm functions to smoothen the effect of term CTR values, so differences between similar ranks become smaller. For this, we integrate natural logarithmic ($\log_e$) and $\log_{10}$ functions [LIV1992].

To see the effect of CTR function, we also use a constant C where it indicates the CTR component weight.

4.2.1 Additive Functions

The general similarity calculation formula for additive approach is as follows.

$$sim(d,q) = \sum_{t \in d,q} \left(w_{tf,d} + R_{tf,d}\right) \cdot \left(w_{tf,q} + R_{tf,q}\right) \cdot w_{idf}$$

For additive alternative, the CTR function can be used as an inverse of its position in document (inverse rank), or inverse of its relative position in document according to document length (percentage rank). The CTR function can also be improved by using logarithmic functions.

If used percentage rank in functions, we may use the maximum or the actual document length. All these functional considerations may depend on each other, so we experiment with all possible combinations given as TABLE 4.1. In all additive formulas, C is a constant generally between 0 and 1 which gives the best experimental results for term rank.

4.2.2 Multiplicative Functions

The general similarity calculation formula for multiplicative approach is as follows.

$$sim(d,q) = \sum_{t \in d,q} \left(w_{tf,d} \cdot R_{tf,d} \right) \cdot \left(w_{tf,q} \cdot R_{tf,q} \right) \cdot w_{idf}$$

For multiplicative alternative, we have two different functions corresponding to inverse and percentage rank. The first method (scaling) scales the CTR score for each term by a value between 1 and C+1, according to percentage rank. The second approach (boosting) scales CTR score in the (1-C, 1] interval. Similar to the additive functions, multiplicative ones can be improved by using classical smoothing logarithmic functions.

Since we use percentage rank in all multiplicative functions, we may use the maximum document length or normal document length alternatives. All these functional considerations may depend on each other similar to additive formulas, so we have to experiment with all these alternatives included in one function. TABLE 4.2 gives all possible functions that can be derived from these considerations. In all multiplicative formulas, C is a constant generally varying between 0 and 1.

TABLE 4.1: Additive function formulas for CTR

Name	dl vs. maxdl[1]	per vs. inv[2]	Logarithm	Function Formula ($R_{tf,d}$)
adp	dl	percentage	-	$C\cdot(1-(tr-1)/dl)$
adpl	dl	percentage	ln	$C\cdot(1-\ln(tr+2)/\ln(dl+2))$
adpl2	dl	percentage	log10	$C\cdot(1-\log_{10}(tr+9)/\log_{10}(dl+9))$
amp	max dl	percentage	-	$C\cdot(1-(tr-1)/\max dl)$
ampl	max dl	percentage	ln	$C\cdot(1-\ln(tr+2)/\ln(\max dl+2))$
ampl2	max dl	percentage	log10	$C\cdot(1-\log_{10}(tr+9)/\log_{10}(\max dl+9))$
ai	-	inverse	-	$C\cdot(1/tr)$
ail	-	inverse	ln	$C\cdot(1/\ln(tr+2))$
ail2	-	inverse	log10	$C\cdot(1/\log_{10}(tr+9)$

TABLE 4.2: Multiplicative function formulas for CTR

Name	dl vs. maxdl[1]	boosted vs. scaled	Logarithm	Function Formula ($R_{tf,d}$)
mdb	dl	boosted	-	$1+C\cdot(1-(tr-1)/dl)$
mdbl	dl	boosted	ln	$1+C\cdot(1-\ln(tr+2)/\ln(dl+2))$
mdbl2	dl	boosted	log10	$1+C\cdot(1-\log_{10}(tr+9)/\log_{10}(dl+9))$
mmb	max dl	boosted	-	$1+C\cdot(1-(tr-1)/\max dl)$
mmbl	max dl	boosted	ln	$1+C\cdot(1-\ln(tr+2)/\ln(\max dl+2))$
mmbl2	max dl	boosted	log10	$1+C\cdot(1-\log_{10}(tr+9)/\log_{10}(\max dl+9))$
mds	dl	scaled	-	$(1-C)+C\cdot(1-(tr-1)/dl)$
mdsl	dl	scaled	ln	$(1-C)+C\cdot(1-\ln(tr+2)/\ln(dl+2))$
mdsl2	dl	scaled	log10	$(1-C)+C\cdot(1-\log_{10}(tr+9)/\log_{10}(dl+9))$
mms	max dl	scaled	-	$(1-C)+C\cdot(1-(tr-1)/\max dl)$
mmsl	max dl	scaled	ln	$(1-C)+C\cdot(1-\ln(tr+2)/\ln(\max dl+2))$
mmsl2	max dl	scaled	log10	$(1-C)+C\cdot(1-\log_{10}(tr+9)/\log_{10}(\max dl+9))$

[1] Document length and maximum document length
[2] Percentage versus inverse rank

Chapter 5

Experimental Design and Evaluation

In this section, we present the experimental findings of chronological term ranking functions given in Chapter 4. For this, we give brief information about our standard TDT collection used throughout this research, then give the evaluation metrics used in these experiments. Lastly, we perform the chronological ranking experiments with additive and multiplicative alternatives.

5.1 Collection Characteristics

Assessing effectiveness of information systems requires a test collection that is suitable for the focused area. The Cranfield experimental evaluation approach with standard test collections has a significant impact on the improvement of IR systems [VOO2007].

Commonly used standard test collections enable objective comparison of different methods aiming at the solution of the same problem. They provide possibility of repeatable evaluations and several baselines for comparison. In TDT, a test collection contains several news articles in temporal order and first stories corresponding to new events and tracking news of a set of events identified by human annotators.

Similar to TDT test collection, our collection contains several topics in temporal order and first stories with tracking events identified by human annotators. The experiments are conducted with this standard TDT collection designed in Turkish language. Some detailed structural information about BilCol2005 is given in TABLE 3.4.

BilCol2005 consists of 80 events, and it is divided to two subsets. The details of training and test sets are given in TABLE 3.3. Also, we have mentioned about the details of training and test sets. BilCol2005 is the first standard test collection in Turkish language, which is constructed in a similar manner to [TDT2004]. In the following experiments, we benefit from both the training and test sets for effectiveness evaluations of NED.

5.2 Evaluation Metrics

New event detection systems performance is measured by two approaches, one using DET curve; other is normalized detection cost function which is explained in a detailed manner in Chapter 3.4. The main sources of these approaches are miss rate and false alarm errors. These error probabilities are combined to a single detection cost by defining the costs of missed detection and false alarms errors a predefined cost values. This approach gives a convenient way of comparison between different systems. As in baseline experiments we benefit from the normalized detection cost defined as follows.

$$C_{\det} = C_{miss} \cdot P_{t\arg et} \cdot P_{miss} + C_{FA} \cdot P_{FA} \cdot (1 - P_{t\arg et})$$

$$(C_{\det})_{Norm} = C_{\det} \Big/ Minimum\{C_{miss} \cdot P_{t\arg et}, C_{FA} \cdot (1 - P_{t\arg et})\}$$

- $C_{miss} = 1$, $C_{FA} = 0.1$ are the costs of a missed detection and a false alarm.
- $P_{target} = 0.02$, the a priori probability of finding a target.
- P_{miss} [1]: miss probability (rate).
- P_{FA} [1]: false alarm probability (rate).

5.3 Baseline Model

We have conducted chronological experiments using the optimum results found in NED experiments. To remind again, we used Okapi similarity function with 50 terms of vector size. We have also used lemmatizer based stemmer and a sliding time-window size of 12 days span. We also used 217 words of stopword list in all of the experiments done in the scope of chronological term ranking.

5.4 Chronological Term Ranking Experiments

In this section, we will give the experimental results of chronological term ranking in first story detection experiments. As explained before, chronological term ranking experiments are divided into two groups: the additive functions and multiplicative functions.

5.4.1 Additive Functions

The additive functions are given in TABLE 4.1. In this part, we conduct two phase experiment sets. In the first phase, we decide the optimum value of C parameter that affects the weight of term ranking in term weighting function. These experiments are

[1] These values are also calculated from topic-weighted method of error probabilities i.e. miss rate and false alarm probabilities.

done with the training set and the results are given in TABLE A.10. According to this experiments C values between [0, 1] with increment value of 0.1 is experimented and for each function the optimum C value is found. Note that the experiments conducted with C values greater than 1 decrease the NED performance, so we do not give the details of these experiments.

TABLE 5.1: Additive formula experiments in training and test sets

Additive Functions	**C**	**Training Set:** $_{\text{norm}}(C_{\text{det}})$	**Improvement (%)**	**Test Set:** $_{\text{norm}}(C_{\text{det}})$	**Improvement (%)**
Baseline	-	0.542	0.000	0.525	0.000
adp	0.8	0.476	13.902	0.512	2.540
adpl	0.8	0.477	13.687	0.514	2.081
adpl2	1.0	0.456	**18.973**	0.521	0.826
amp	0.4	0.539	0.575	0.527	-0.398
ampl	0.3	0.508	6.793	0.509	3.225
ampl2	0.4	0.507	6.961	0.503	4.395
ai	0.6	0.489	10.966	0.509	3.043
ail	0.8	0.490	10.784	0.480	**9.468**
ail2	0.3	0.508	6.793	0.509	3.063

After optimizing the function parameter for each experiment, we conduct the experiments with test set using optimum C values. As seen from TABLE 5.1, additive CTR functions generally perform better than the baseline system. If we examine the results of additive functions, the performance of percentage function with document length generally performs better in the training set, but it does not continue the similar performance benefit in the test set. The only function that maintains the performance gain in both systems is additive inverse function with natural logarithm (ail). The performance gain with this additive function is about 10%.

5.4.2 Multiplicative Functions

The multiplicative functions are given in TABLE 4.2. In this section, we have conducted a two stage experiment similar to the additive experiments. In the first stage, we have evaluated the optimum value of C parameter that affects the weight of term ranking in term weighting function. These experiments are done with the training set and the outcomes are given in TABLE A.11. According to the experiments C values between [0, 1] with increment value of 0.1 are experimented and for each function the optimum C value is obtained. Note that the experiments conducted with C values greater than 1 decrease the NED performance, so we do not give the details of these experiments.

TABLE 5.2: Multiplicative formulas experiments in training and test Sets

Multiplicative Functions	C	Training Set $_{norm}(C_{det})$	Improvement (%)	Test Set $_{norm}(C_{det})$	Improvement (%)
Baseline	-	0.542	0.000	0.525	0.000
mdb	0.3	0.479	13.236	0.510	3.023
mdbl	0.4	0.477	13.782	0.466	**12.567**
mdbl2	0.5	0.458	**18.402**	0.525	0.019
mmb	0.5	0.517	4.954	0.506	3.797
mmbl	0.3	0.504	7.619	0.494	6.212
mmbl2	0.5	0.503	7.747	0.493	6.449
mds	0.2	0.489	10.966	0.498	5.486
mdsl	0.2	0.493	9.953	0.524	0.114
mdsl2	0.4	0.475	14.117	0.531	-1.186
mms	0.7	0.521	4.028	0.499	5.296
mmsl	0.1	0.511	6.249	0.503	4.333
mmsl2	0.3	0.509	6.541	0.514	2.041

After determining the function parameter C, we execute the experiments with the test set. As seen from TABLE 5.2, we conduct two sets of experiments which are boosted multiplicative and scaled multiplicative function sets. According to results the best performance is gained in the training set with multiplicative boost function using log10 and document length (mdbl2) with improvement about 18%. However, the same performance gain has not achieved in the test set. Some of the functions maintain the performance gain in both training and test sets. These functions are mdbl (boosted multiplicative using natural logarithm and document length), mmbl (boosted multiplicative using natural logarithm and maximum document length), mmbl2 (boosted multiplicative using log10 and maximum document length), mds (scaled multiplicative with document length), and mms (scaled multiplicative with maximum document length). The best performance gain among these functions is depicted as mdbl with about 13% percentage success in both training and test sets.

5.5 Chapter Summary

In this chapter, we show that using chronological term ranking approach with different parameters and formulas improves the NED performance. We divide the chronological term ranking experiments into two sets of experiments: using additive functions as CTR formula, using multiplicative function as CTR value. In each experiment we have determined the parameter values (C) in CTR function formulas using the training set. In the second part, using the optimum values of C, we conduct experiments using the test set. In nearly all experiments, the NED performance increases. The best results are gathered ail (additive inverse function with natural logarithm) using additive approach and mdbl (boosted multiplicative using natural logarithm and document length) using multiplicative approach. The performance gains with these additive functions are about more than 10%.

The observations of the experimental results are as follows.

- The chronological term positions in documents are important for NED experiments. Because general event detection systems are fed from news collections, which are written with pyramid style.
- The short documents should be handled differently in evaluation of chronological term ranking. The success of maximum document length points that the CTR values of short documents may be lost using only document length in functions. Also in additive functions ail shows that the relative document length is not important, the important thing is that the term position.

These experiments provide a good foundation for future work in the enhancement of other chronological term ranking functions for NED.

Chapter 6

Further Experiments and Discussion

In the previous experiments, we have explored the use of chronological term ranking functions in term weighting in the scope of new event detection. We have experimentally shown that chronological term position for news texts gives important clues of document. In order to make our experimental findings more robust among different conditions, we enhance the variety of our experiments done with CTR functions. In this section we firstly talk about our enhancements in experimental design, and then we also discuss some other approaches that might enhance CTR performance.

6.1 N-Pass Detection Experiments

In Chapter 5, we have conducted experiments with one data set and make some conclusions according to these results. To improve the reliability, we vary our dataset by using a known technique, N-pass detection evaluation which generates N datasets from one dataset. After each pass, the first story of each event is removed, and detection and evaluation are applied again to the corpus [YAN1998].

In our case dataset we conducted a 6-pass detection evaluation. The six passes are labeled by N skip = 0, l, 2, 3, 4, 5. The intuition behind using 6 pass is from TABLE A.1. In this table, the details of event clusters are given and the minimum number of tracking stories is 5. The results of additive and multiplicative functions are given in TABLE 6.1 and TABLE 6.2.

TABLE 6.1: Additive function performance with six-pass detection

Additive Functions	**N=0**	**N=1**	**N=2**	**N=3**	**N=4**	**N=5**	**Average**	**Improvement (%)**	**p value**
Baseline	0.525	0.599	0.508	0.581	0.523	0.685	0.570	-	-
adp	0.512	0.572	0.492	0.526	0.534	0.685	0.553	3.045	0.065[1]
adpl	0.514	0.565	0.474	0.528	0.520	0.666	0.545	4.685	0.009[2]
adpl2	0.521	0.582	0.492	0.541	0.539	0.678	0.559	2.071	0.091[1]
amp	0.527	0.604	0.508	0.575	0.507	0.684	0.567	0.476	0.208
ampl	0.509	0.589	0.511	0.535	0.508	0.671	0.553	3.014	0.026[2]
ampl2	0.503	0.586	0.511	0.541	0.508	0.671	0.553	3.070	0.015[2]
ai	0.509	0.527	0.507	0.542	0.519	0.653	0.543	5.007	0.027[2]
ail	0.480	0.524	0.478	0.511	0.510	0.641	0.524	**8.821**	0.002[2]
ail2	0.509	0.586	0.511	0.535	0.507	0.670	0.553	3.114	0.023[2]

As seen from the results, ail (additive inverse function with logarithm) performs the best among all results calculated as the average of five passes. Also mdbl (boosted multiplicative using natural logarithm and document length) also performs compatible to ail and better than all others. These experimental observations assert that the chronological position information is an important feature in NED experiments. The predefined CTR functions generally give better results than the baseline approach. We also perform one-tailed t-tests (matches pair) with the performances against baseline. One sided p values are given in TABLE 6.1 and TABLE 6.2. According to p values, all of the additive experiments indicate that chronological term ranking based NED is

[1] Nearly Significant

[2] Strongly Significant

nearly significantly (p<0.1) or strongly significantly (p<0.05) different. The similar results are achieved with multiplicative approach where all except one is either nearly or significantly different from baseline approach. These results support that, chronological term ranking approach gives statistically significantly different results against the baseline approach.

TABLE 6.2: Multiplicative function performance with six-pass detection

Multiplicative Functions	N=0	N=1	N=2	N=3	N=4	N=5	Average	Improvement (%)	p value
Baseline	0.525	0.599	0.508	0.581	0.523	0.685	0.570	-	-
mdb	0.510	0.568	0.478	0.526	0.537	0.669	0.548	4.086	0.031^2
mdbl	0.466	0.560	0.478	0.527	0.520	0.658	0.535	**6.604**	0.004^2
mdbl2	0.525	0.572	0.482	0.537	0.539	0.671	0.554	2.866	0.064^1
mmb	0.506	0.587	0.511	0.575	0.513	0.676	0.561	1.607	0.015^2
mmbl	0.494	0.563	0.486	0.531	0.511	0.647	0.539	5.871	0.001^2
mmbl2	0.493	0.562	0.486	0.515	0.515	0.642	0.535	6.484	0.004^2
mds	0.498	0.564	0.486	0.557	0.525	0.678	0.551	3.416	0.010^2
mdsl	0.524	0.577	0.487	0.531	0.524	0.676	0.553	3.080	0.038^2
mdsl2	0.531	0.596	0.484	0.553	0.544	0.676	0.564	1.061	0.230
mms	0.499	0.591	0.512	0.582	0.518	0.651	0.559	2.022	0.068^1
mmsl	0.503	0.588	0.512	0.579	0.523	0.675	0.563	1.234	0.064^1
mmsl2	0.514	0.571	0.510	0.543	0.528	0.644	0.552	3.341	0.037^2

6.2 Performance Comparison of CTR Functions

In this part, we perform some statistical tests with the results of N-Pass detection experiments. We have conducted statistical tests using pair-wise comparisons between chronological ranking functions. The statistical tests are conducted in two sets of experiments. We prefer one sided matched pair t-tests with multiplicative functions and additive functions. The p values are given in TABLE A.12 and TABLE A.13. The intuition behind pair-wise comparison is to select the best chronological term ranking combination with NED systems. We conclude that almost all of the functions are significantly different. According to pair-wise results we conclude that ail is significantly different from all functions which performs the best among additive functions. Also, for multiplicative functions mdbl, mmbl, and mmbl2 are not statistically different so we can use either of them in multiplicative functions. When we compare with ail and other multiplicative functions; ail is significantly different from mmbl and mmbl2. Also, ail is nearly significantly different from mdbl. So, from the statistical results we conclude that the best chronological term ranking function is ail which performs the best among all chronological term ranking functions.

6.3 Future Development Possibilities

In this research, the CTR functions are incorporated with classical term weight using multiplicative and additive functions. But there may be many other methodologies. One approach may be using the sentence level chronological term positioning approach. This approach assumes that the position of term is evaluated by the sentence position of the earliest term occurrence. Another approach may be evaluating the similarity of documents using only CTR weighting and incorporating the CTR similarity with classical similarity by using some fusion methods. Also as we have implied before, we have only formed the baseline for CTR functions. It is an open challenge of finding different CTR approaches to improve the relevance performance between documents.

Chapter 7

Conclusions and Future Work

One of the challenging tasks coming with intelligent news portals is new event detection. It aims to find the novel stories coming in a news stream. In this thesis, we study the new event detection (NED) problem. We analyze the performance of NED in Turkish language and propose some novel solutions to increase effectiveness of the first story detection. We extend the previous works in the new event detection systems by using chronological term ranking approach. The experimental results show that our approach outperforms a baseline system with a desirable performance.

7.1 Thesis Summary

As the baseline NED system, we use the Okapi similarity measure with top 50 terms obtained by using a lemmatizer. After defining the baseline for NED system, we focus on some intrinsic features that might increase the effectiveness in NED. For this purpose, we examine the importance of term ranking. This feature is generally not used for similarity calculations and may be beneficial for NED experiments which generally deal with news articles where chronological ranking gives information about the document characteristics. We propose several functions for enhancing the relevance scoring for NED experiments.

In the experiments we analyzed several chronological term ranking functions with different parameters and formulas. The results show that NED performance increases using this CTR approach. The performance gains with some of the functions provide up to 13% improvement. It is desirable performance gain for the first story detection systems.

Lastly, to generalize and measure the robustness of our chronological ranking approach, we conducted N-pass detection experiments with N equal to 6. This approach enriches the experimental dataset by using incremental removal process in the first N stories of the event cluster. Then the evaluation is carried out with N different dataset. The performances of CTR functions are statistically significantly more effective than the baseline performance. Further statistical experiments show that ails (additive inverse function with logarithm) outperforms all of the other CTR functions.

7.2 Contributions and Future Work

In this thesis, we propose some changes in the term weighting component of Okapi by incorporating the chronological term ranking information. The results show that chronological term ranking functions improve the effectiveness of first story detection systems. We also conduct experiments with scaling our dataset using N-pass detection. The results show that our approach has a robust performance gain against the baseline approach.

We extend the previous works in NED using CTR for term weighting functions. The major contributions of this work are the following.

- Various experiments are conducted with Turkish TDT dataset. We have conducted experiments with various stemming options and similarity functions.
- With the intuition that the news articles are written using reverse pyramid model, we have devised various chronological term ranking functions. We have experimentally shown that term position information is an important intrinsic feature that can be used for term weighting in new event detection systems when the application area is news articles.
- We have formed the baseline chronological term ranking functions that can be used in NED experiments. We also validated the robustness of CTR functions using further experiments (6-pass detection).

The research described in this thesis can be extended in many directions. We can

- Introduce new chronological term ranking functions using sentence position of the earliest term occurrence.
- Evaluate the similarity of documents using only CTR weighting and incorporating the CTR similarity with classical similarity by using some data fusion methods.
- Incorporate chronological term ranking approach with other systems such as event tracking, story link detection, information filtering, event summarization, and news copy detection to improve their respective effectiveness.
- Extend chronological term ranking functions using different mathematical approaches.

References

[ALL1996] J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 270-278, 1996.

[ALL1999] J. Allan, H. Jin, M. Rajman, C. Wayne, D. Gildea, V. Lavrenko, R. Hoberman, D. Caputo. Topic-based novelty detection. In *1999 summer workshop at CLSP, final report*, 1999.

[ALL2000] J. Allan, V. Lavrenko, H. Jin. First story detection in TDT is hard. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, pp. 374-381, 2000.

[ALL2002] J. Allan (Ed.), V. Lavrenko, R. Swan. Explorations within topic tracking and detection. *Topic Detection and Tracking. Event-Based Information Organization*, pp. 197-224, Kluwer Academic Publishers, 2002.

[ALT2007] K. Altintas, F. Can, J. M. Patton. Language change quantification by time-separated parallel translations. *Literary and Linguistis Computing*, Vol. 22, No. 4, pp. 375-393, 2007.

[AMI2007] L. Amini. Stream processing: What's in it for you? IBM T. J. Watson Research Center, Sep. 12, 2007. Retrieved March 8, 2008, from http:// www-05.ibm.com/nl/events/presentations/stream_processing_whats_in _it_for_you.pdf. 2007.

[AND1973] M. R. Anderberg. *Cluster Analysis for Applications*. New York: Academic Press, 1973.

[BAE1999] R. Baeza-Yates and B. Riberio-Neto. *Modern Information Retrieval*. New York: ACM Press, 1999.

[BAG1993] N. Bagnall. *Journalism Media Manual: Newspaper Language*. Media Manuals. Focal Press, 1993.

[BEL1998] A. Bell and P. Garrett. Approaches to Media Discourse. Willey Blackwell, 1998.

[BIL2009] P. Bill. Basic news writing. Retrieved March 8, 2009. Ohnole College. Journalism Department, 2009.

[BRA2003] T. Brants, F. Chen, A. Farahat. A system for new event detection. *In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 330-337, 2003.

[CAN2008a] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, O. M. Vursavas. Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology*, Vol. 59, No. 2. 407-421. 2008

[CAN2008b] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H.C. Ocalan, E. Uyar. Bilkent News Portal: A personalizable system with new event detection and tracking capabilities. *The 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. p. 885, 2008.

[CAN2009] F. Can, S. Koçberber, Ö. Bağlıoğlu, S. Kardaş, H. C. Öcalan, E. Uyar. Topic detection and tracking in Turkish. *Journal of the American Society for Information Science and Technology* (submitted).

[DOD1997] M. A. Doddington, G. Kamm, M. Ordowski, M. Pryzybocki. The DET curve assesment of detection task performance. *In Proceeding of Eurospeech 97'*, Vol. 4, pp 35-46. 1997.

[DOD1998] G. Doddington. The topic detection and tracking phase 2 evaluation plan. http://www.nist.gov/speech/tdt98/doc/tdt2.eval.plan.98.v3.7.pdf. 1998.

[DOD1999] G. Doddington. Presentation slides. *The Darpa Broadcast News Workshop*. Herndon, VA, 1999.

[FIS2002] J. G. Fiscus and G. R. Doddington. Topic detection and tracking evaluation overview. In J. Allan (Ed.), Topic Detection and Tracking Event-based Information Organization. pp. 17-31, Norwell, MA: Kluwer Academic Publishers. 2002.

[FIS2004] J. Fiscus and B. Wheatley. Overview of the TDT 2004 evaluation and results. Retrieved July 15, 2007, from http://www.nist.govs/speech/tests/tdt/tdt2004/papers/NIST-TDT2004.ppt. 2004.

[FLY2009] W. Flyer. Wright-ing prompt: News article, language arts. Retrieved March 8, 2009 from http://quest.arc.nasa.gov/aero/wright/teachers/pdf/language/Newspaper_Article.pdf. 2009.

[FRA2001] M Franz, W. Todd, J. S. McCarley, Z. Wei-Jing. Unsupervised and supervised clustering for topic tracking. *The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 310-317, 2001.

[IRV1991] F. Irving. Writing style differences in newspaper, radio, and television news. University of Minnesota. Monograph Series.. No. 2, pp.3-5, 1991.

[JIN1999] H. Jin, R. Schwartz, S. Sista, F. Walls. Topic tracking for radio, TV, broadcast and newswire. *In Proceedings of the DARPA Broadcast News Workshop 99'*, pp 199-204, 1999.

[KAR2009] S. Kardaş. *New Event Detection and Tracking in Turkish*. Master Thesis. Computer Engineering Department, Bilkent University, 2009.

[KEN2009] Ken Blake. Inverted pyramid story format. Retrieved 1 May 2008 from http://kelab.tamu.edu/spb_encyclopedia/data/Inverted%20pyramid%20story%20format.pdf. Middle Tennessee State University, 2009.

[KUM2004a] G. Kumaran and J. Allan. Text classification and named entities for new event detection. *In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 297-304, 2004.

[KUM2004b] G. Kumaran, J. Allan, A. McCallum. Classification models for new event detection. *In Proceedings of 13th Annual International Conference on Information Knowledge Management*. 2004.

[KUM2005] Giridhar Kumaran, James Allan .Using names and topics for new event detection. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language*, pp. 121-128, 2005.

[KUR2001] H. Kurt. *On-Line New Event Detection and Tracking in A Multi-Resource Environment*. Master Thesis, Computer Engineering Department, Bilkent University, 2001.

[LIV1992] S. A. LivingStone. Small sample equating with log-linear smoothing. Educational Testing Service. Princeton Newyork. 1992.

[LUO2007] C. T. Luo, P. S. Yu. Resource-adaptive new event detection. *In Proceedings of the 27th International Conference on Management of Data*, pp. 497-508, 2007.

[MAK2003] J. Makkonen. Investigation of Event Evaluation in TDT. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language,* pp: 43 – 48, 2003.

[MAK2004] J. Makkonen, H. Ahonen-Myka, M. Salmenkivi. Simple semantics in topic detection and tracking. *Information Retrieval*. Kluwer Academic Publishers Vol. 7, No. 3-4 pp. 347-368, 2004.

[MIT2006] Mitchell Stephens. *A History of News*. Oxford University Press. 2006.

[OCA2009] H. Ç. Öcalan. *Bilkent News Portal: A System with New Event Detection and Tracking Capabilities*. Master Thesis. Computer Engineering Department, Bilkent University, 2009.

[OFL1994] K. Oflazer. Two-level description of Turkish morphology, *Literary and Linguistic Computing*. Vol. 9 n. 2, pp. 137-148, 1994.

[PAP1999] R. Papka. O*nline Event Detection, Clustering and Tracking*. Phd Disertation Thesis. University of Massachusetts at Amherst,1999.

[PAP2000] R. Papka and J. Allan. Topic Detection and Tracking: Event Clustering as a Basis for First Story Detection. Advances Information Retrieval: Recent Research from the CIIR, W. Bruce Croft, ed. cp. 4, pp. 96-126, 2000.

[RIJ1979] van Rijbergen C.J. *Information Retrieval*. Butterworths, London, 1979.

[SAL1975] G. Salton, A. Wong, C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, vol. 18 n. 11, p. 613-620, 1975.

[SAL1988] G. Salton, C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Process and Management*, vol 24 n. 5 pp. 513-523, 1988.

[SIN2001] A. Singhal. Modern Information Retrieval: A Brief Overview. Google Inc. *IEEE Computer Society Technical Committee on Data Engineering*, vol. 24, n. 4, pp. 35-43, 2001.

[STO2001] N. Stokes, J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection [Poster paper]. *In Proceedings* *of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 424- 425, 2001.

[TDT2004] TDT annotation manual: Version 1.2 – August 4, 2004. Retrieved January 9, 2007, from http://projects.ldc.upenn.edu/TDT5/Annotation/TDT2004V1.2.pdf, 2004

[TDT2008] Topic detection and tracking evaluation. Retrieved June 18, 2008, from. http://www.itl.nist.gov/iaui/894.01/tests/tdt/, 2008.

[TRO2007] A. D. Troy and G. Zhang. Enhancing relevance scoring with chronological term rank. *In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 599-606, 2007.

[UNV1996] N. Unwin. The individuation of events. *Mind*, vol. 105 n.418, pp. 315-330, 1996.

[UYA2009] E. Uyar. *Near Duplicate News Detection Using Named Entities. Master Thesis*. Computer Engineering Department, Bilkent University, 2009.

[VAR2005] H. R. Varian. Universal Access to Information. *The Digital Society*, vol 48, i. 10, pp: 65 – 66, 2005.

[VOR2007] E. M. Voorhees. TREC: Continuing information retrieval's tradition of experimentation. *Communications of the ACM*, vol. 50, n. 11, pp. 51-54, 2007.

[YAN1998] Y. Yang, T. Pierce, J. Carbonell. A Study on retrospective and on-line event detection. *In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 28-36, 1998.

[YAN2002] Y. Yang, J. Carbonell, R. Brown, J. Lafferty, T. Pierce, T. Ault. Multi-strategy learning for topic detection and tracking. In J. Allan (Ed.), Topic

Detection and Tracking Event-based Information Organization: pp. 85-114, 2002.

[ZHA2007] K. Zhang, J. Zi, L. G. Wu, L. New event detection based on indexing-tree and named entity. *In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 215-222 2007.

Appendices

Appendix A: Information for Annotated News

TABLE A.1:Summary information for annotated news

Event No	Brief Description of Event	Number of Tracking Stories	Time Span (Days)	Event Days (Month/Day)
1	Kars'ta Trafik Kazası 7 öldü	20	203	05/28 - 12/16
2	Onur Air'in Hollanda’ya inişi yasaklandı	159	203	05/12 - 11/30
3	Koreli bilim adamının kök hücre araştırması sahte	8	11	12/19 - 12/29
4	Nema karşılığı kredi	31	280	02/08 - 11/14
5	Tokyo'da trenlerde haremlik selamlık	8	263	04/04 - 12/22
6	Londra metrosunda patlama	454	175	07/07 - 12/28
7	Barbaros Çocuk Köyü'nde çocuk tacizi skandalı	88	275	01/26 - 10/27
8	Formula G	20	58	07/04 - 08/30
9	Karamürsel kaymakamı intihar etti	6	7	01/04 - 01/10
10	400 koyun intihar etti	10	8	07/08 - 07/15
11	Şemdinli olayları	317	53	11/09 - 12/31
12	Türkiye'de kuş gribi	229	83	10/10 - 12/31
13	Şampiyon Fenerbahçe	115	222	05/22 - 12/29
14	Mortgage Türkiye'de	375	357	01/07 - 12/29
15	2005 Avrupa Basketbol Şampiyonası	78	297	01/15 - 11/07
16	Yüzüncü Yıl Üniversitesinde ihale yolsuzluğu iddiası	326	79	10/14 - 12/31
17	Kral Fahd hastaneye kaldırıldı	51	77	5/27 - 08/11
18	Memurlarının bir üst dereceye terfisi	52	110	01/6 - 04/25

Event No	Brief Description of Event	Number of Tracking Stories	Time Span (Days)	Event Days (Month/Day)
19	Bill Gates Türkiye'ye geldi	17	8	01/30 - 02/06
20	Mısır'da patlamalarda çok sayıda kişi öldü	120	43	07/23 - 09/03
21	Atillâ İlhan vefat etti	40	70	10/11 - 12/19
22	Ata Türk'ün ölümü	43	47	09/18 - 11/03
23	DT Genel Müdürü Lemi Bilgin görevden alındı	63	109	08/19 - 12/05
24	Universiade 2005 Yaz Spor Oyunları	248	289	03/04 - 12/17
25	Yahya Murat Demirel Bulgaristan'da yakalandı	192	345	01/03 - 12/13
26	Bağdat El Ayma köprüsünde izdiham	29	9	08/31 - 09/08
27	Prof. Dr. Sadettin Güner'e saldırı	41	291	01/08 - 10/25
28	Nestle'de mürekkepli süt	11	2	11/22 - 11/23
29	Nermin Erbakan tedavi altında	45	46	10/20 - 12/04
30	Ulubey'de çocukla annenin peş peşe ölümü	6	31	05/19 - 06/18
31	15. Akdeniz Oyunları	193	86	05/02 - 07/26
32	Kemal Derviş'in UNDP başkanı seçilmesi	118	181	03/11 - 09/07
33	Irak başbakanı Caferi Tahran'ı ziyaret etti	22	94	07/05 - 10/06
34	Gediz'de grizu patlaması	39	36	04/21 - 05/26
35	Sarıgül'ün CHP'de kendini savunması	110	352	01/02 - 12/19
36	Paris'de polisle göçmenler arasındaki çatışma	245	51	10/29 - 12/18
37	Rock'n Coke açık hava müzik etkinliği	11	5	09/02 - 09/06
38	Ankara Garı'nda tren kazası	13	5	01/13 - 01/17
39	2005 Nobel tıp ödülü	19	75	10/03 - 12/16
40	Kayseri Erciyes Üniversitesindeki bebek ölümleri	39	60	08/03 - 10/01
41	Marburg virüsünden ölenler	25	65	03/16 - 05/19
42	Gamze Özçelik'in görüntüleri	43	116	08/29 - 12/22
43	Türkiye'nin ilk yediz bebekleri geliyor	56	301	02/17 - 12/14
44	Yeni Türk Ceza Kanunu	53	193	06/01 - 12/10
45	Saddam Hüseyin'in yargılanmasına başlandı	182	72	10/19 - 12/29
46	Beylikdüzü çöpte patlama	17	5	11/18 - 11/22
47	Endonezya'nın Bali Adası'nda 4 bomba patladı	15	4	10/01 - 10/04
48	Sahte rakı	323	182	03/01 - 08/29
49	Hindistan'da üç saldırıda 66 kişi öldü	21	5	10/29 - 11/02
50	Bülent Ersoy ve Deniz Baykal polemiği	52	132	08/19 - 12/28
51	Tahran'da askeri uçak düştü	9	2	12/06 - 12/07
52	Sochi seferinde Ufuk-1 gemisi yanmaya başladı	20	3	08/25 - 08/27
53	İstanbul'da kanalizasyonda işçiler zehirlendi	9	2	12/05 - 12/06

Event No	Brief Description of Event	Number of Tracking Stories	Time Span (Days)	Event Days (Month/Day)
54	Kadınlara copla müdahale eden polisler	104	297	03/06 - 12/27
55	Kuşadası'nda minibüsteki patlama	50	4	07/16 - 07/19
56	Esenboğa Havalimanı iç hatlarda yangın	18	36	11/14 - 12/19
57	Zeytinburnu'nda bir evde patlama	28	4	08/08 - 08/11
58	Malatya çocuk yuvasında işkence	192	67	10/26 - 12/31
59	ABD denizaltısı ile Türk gemisi çarpıştı	7	1	09/05 - 09/05
60	Prof Dr. Kalaycı suikast sonucu öldürüldü	44	23	11/11 - 12/03
61	İlk yüz nakli	14	17	12/01 - 12/17
62	15 yeni üniversite kurulmasına ilişkin kanun	59	50	11/12 - 12/31
63	Gaziantep tanker patlaması	33	7	08/06 - 08/12
64	Hakkâri'de bomba patladı	10	4	07/29 - 08/01
65	Erzurum çocuk yuvasında bebek ölümlü	9	3	11/04 - 11/06
66	Kâzım Koyuncunun ölümü	30	129	06/25 - 10/31
67	Melih Kibar'ın ölümü	16	120	04/07 - 08/04
68	Sarıkamış şehitleri anıldı	5	3	12/23 - 12/25
69	Endonezya'da yolcu uçağı düştü	15	2	09/05 - 09/06
70	Şanlıurfa'da köprü inşaatı çöktü	7	2	04/13 - 04/14
71	Japonya Osaka'da tren kazası	29	4	04/25 - 04/28
72	Manken Tuğçe Kazaz'ın hıristiyan oldu	11	76	09/22 - 12/06
73	Fotoğraf sanatçısı Mehmet Gülbiz öldürüldü	14	127	02/04 - 06/10
74	Atina'daki Kara Harp Okulu'nda Türk bayrağı olayı	55	71	04/16 - 06/25
75	Maslak'ta patlama	30	18	10/15 - 11/01
76	Didim'de denize uçak düştü	13	2	07/19 - 07/20
77	Rum yolcu uçağı düştü	106	115	08/14 - 12/06
78	İstiklal Caddesindeki ağaçlar kaldırıldı	8	16	11/02 - 11/17
79	Zeytinburnu'nda gemi battı	38	3	03/13 - 03/15
80	İngiltere'de Osmanlı kültürü hakkında sergi açıldı	22	103	01/01 - 04/13
Avg.	-	73	92	-
Min.	-	5	1	-
Max.	-	454	357	-

Appendix B: Stopword List

TABLE A.2: Stopword list (217 words)

acaba	böylece	ediliyor	içinse	nedeni	olsa	şöyle
ama	böylesi	edilmesi	ile	nedenle	olsaydı	şöyleydi
ancak	bu	ediyor	ilgili	nedense	olsun	şu
arada	budur	eğer	ise	neler	olup	şunlar
ayrıca	buna	etme	işte	niye	olur	şunları
bana	bundan	etmesi	itibaren	o	olursa	tarafından
bazen	bunlar	etmeye	itibariyle	olan	oluyor	üstelik
bazı	bunları	etmişti	kadar	olarak	ona	üzere
bazıları	bunların	etti	karşın	oldu	onlar	var
bazısı	bunu	ettiği	kendi	olduğu	onlara	vardı
belki	bunun	ettiğinde	kendileri	olduğunda	onları	varmış
ben	burada	ettiğine	kendilerine	olduğunu	onların	ve
bence	çok	ettiğini	kendine	oldukça	onu	veya
beni	çünkü	gibi	kendini	oldukları	onun	ya
benim	da	gibidir	kendisi	olduklarını	oysa	yalnızca
beri	daha	gibiydi	kendisine	olduysa	öyle	yani
bile	dahası	göre	kendisini	olma	öylesi	yapacak
bir	de	halen	kendisinin	olmadan	öyleyse	yapılan
birçoğu	değil	hangi	ki	olmadı	pek	yapılması
birçok	değildi	hangisi	kim	olmadığı	peki	yapıyor
biri	değilmiş	hatta	kimse	olmak	rağmen	yapma
birkaç	diğer	hem	kimsenin	olması	sadece	yapmak
birkaçı	diğeri	henüz	kimseye	olmasın	sanki	yapması
biz	diye	her	mı	olmasına	sen	yaptı
bizce	dolayı	herhangi	mi	olmasını	senin	yaptığı
bize	dolayısıyla	herkesçe	mu	olmayan	siz	yaptığını
bizi	edecek	herkesin	mü	olmayıp	sizin	yaptıkları
bizim	eden	hiç	nasıl	olmaz	şey	yerine
bizimdir	ederek	hiçbir	nasılsa	olmuş	şeyden	yine
bizimki	ederse	için	ne	olmuşsa	şeyi	yoksa
böyle	edilecek	içindi	neden	olmuştu	şeyler	zaten

Appendix C: Topic Weighted vs. Story Weighted Evaluation

Topic Weighted and Story Weighted Calculation of NED Error Probabilities

Miss and false alarm probabilities are the primary measures used to represent system performance in the Topic Detection and Tracking (TDT) program. These error probabilities are estimated over an evaluation data set that comprises a large number of stories and a modest number of topics. The usual method of estimating error probabilities is to pool all decisions:

$$\mathbf{P_{Miss}} = \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ \left(1 - \delta_{hyp}(t,s)\right) \cdot \delta_{ref}(t,s) \right\} \right\} \Big/ \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \delta_{ref}(t,s) \right\}$$

$$\mathbf{P_{FalseAlarm}} = \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ \delta_{hyp}(t,s) \cdot (1 - \delta_{ref}(t,s)) \right\} \right\} \Big/ \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left(1 - \delta_{ref}(t,s)\right) \right\}$$

Where,

$$\delta_{sys}(t,s) = \begin{cases} 1 & \text{if } sys \text{ deemed that topic } t \text{ was discussed in story } s \\ 0 & \text{otherwise} \end{cases}$$

$$Stories_t = \text{all stories in the test corpus after the last training story for topic } t$$

This method is called ***story-weighted*** because each story contributes equally to the error estimates. However, because error probabilities are strongly dependent on topic, because there are only a modest number of topics, and because the number of stories per topic varies greatly, it may be desirable to give each topic equal weight:

$$\mathbf{P}_{\mathbf{Miss}} = \frac{1}{N_{Topics}} \sum_{t\in Topics} \left\{ \sum_{s\in Stories_t} \{(1-\delta_{hyp}(t,s))\cdot \delta_{ref}(t,s)\} \Big/ \sum_{s\in Stories_t} \delta_{ref}(t,s) \right\}$$

$$\mathbf{P}_{\mathbf{FalseAlarm}} = \frac{1}{N_{Topics}} \sum_{t\in Topics} \left\{ \sum_{s\in Stories_t} \{\delta_{hyp}(t,s)\cdot (1-\delta_{ref}(t,s))\} \Big/ \sum_{s\in Stories_t} (1-\delta_{ref}(t,s)) \right\}$$

This method is called ***topic-weighted*** because each topic contributes equally to the error estimates [DOD1998].

Here is an example that makes story weighted and topic weighted evaluation of TDT more clearly understandable.

Example

Think that we have four topics in our corpus for evaluation as seen in. For new event detection, let's compute the false probabilities and miss rates with story weighted and topic weighted approaches.

TABLE A.3: Error probability calculation example

Topic Number	# of Tracking News	New Event Detection	False Alarm
1	40	Hit	2
2	30	Miss	3
3	20	Hit	4
4	10	Miss	1

In first story detection, suppose that we missed the second and the fourth topics' seeds. So, in story weighted approach according to the formula;

$$P_{miss} = \frac{((1-1)\cdot 1) + ((1-0)\cdot 1) + ((1-1)\cdot 1) + (1-0)\cdot 1}{((1+39\cdot 0) + 1 + 29\cdot 0) + (1+19\cdot 0) + (1+9\cdot 0)}$$

$$P_{miss} = 2/4$$

Where δ_{ref} (t,s)=1 if the story is the first story, else 0.

In topic weighted approach, δ_{ref} (t,s) of stories doesn't change. According to the formula;

$$P_{miss} = \frac{1}{4}\cdot(\frac{(1-1)\cdot 1}{(1+40\cdot 0)} + \frac{(1-0)\cdot 1}{(1+30\cdot 0)} + \frac{(1-1)\cdot 1}{(1+20\cdot 0)} + \frac{(1-0)\cdot 1}{(1+10\cdot 0)})$$

$$P_{miss} = 2/4$$

To find false alarm probabilities for this example according to the formula again with story weighted approach;

$$P_{fa} = \frac{2\cdot(1\cdot(1-0)) + 3\cdot(1\cdot(1-0)) + 4\cdot(1\cdot(1-0)) + 1\cdot(1\cdot(1-0))}{40\cdot(1-0) + 30\cdot(1-0) + 20\cdot(1-0) + 10\cdot(1-0)}$$

$$P_{fa} = 10/100$$

The false alarm probabilities can also be found by using topic weighted approach;

$$P_{fa} = \frac{1}{4}\cdot\left(\frac{2\cdot(1\cdot(1-0))}{40\cdot(1-0)} + \frac{3\cdot(1\cdot(1-0))}{30\cdot(1-0)} + \frac{4\cdot(1\cdot(1-0))}{20\cdot(1-0)} + \frac{1\cdot(1\cdot(1-0))}{10\cdot(1-0)}\right)$$

$$P_{fa} = 9/80 = 0.1125$$

For first story detection as seen from the examples, story weighted and topic weighted approaches in false alarms probabilities, but they will be same in miss probabilities. This is obvious in FSD systems that miss rates does not change with story weighted or topic weighted approaches, however false alarm rates differ between different approaches.

Appendix D: Similarity Function Selection Experiments

TABLE A.4: Okapi similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.8356	0.7346	0.6590	**0.6002**	0.6075	0.6104	0.6393	0.6496	0.6591	0.6707	0.7093	0.6993	0.6767	0.7225	0.7200	0.7406
F6	0.7730	0.7470	0.6969	0.6859	0.6682	0.6565	**0.6299**	0.6467	0.6335	0.6596	0.7054	0.6963	0.7161	0.7348	0.7258	0.7458
LM	0.7654	0.6376	0.6120	0.6271	**0.5424**	0.6069	0.5900	0.6084	0.6225	0.6446	0.6813	0.6911	0.6971	0.7135	0.7116	0.7194
NS	0.9395	0.7569	0.7013	0.7737	0.7573	0.7252	0.7270	0.6873	0.6917	**0.6314**	0.6638	0.6823	0.7046	0.7123	0.7170	0.7317

TABLE A.5: Jaccard similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.7846	0.6950	0.6196	0.6279	0.6637	0.6761	0.663	0.6452	0.6444	0.6403	0.6359	0.6363	0.6422	0.6147	0.6079	**0.5781**
F6	0.7498	0.7436	0.7261	0.6582	0.7002	0.7075	0.6982	0.6876	0.6761	0.6764	0.6703	0.6613	0.6497	0.6449	0.6449	**0.6139**
LM	0.7683	0.6246	0.6290	0.6241	0.6070	0.5947	0.6216	0.6042	0.5963	0.5953	0.6037	0.5902	0.5860	0.6105	0.6048	**0.5664**
NS	0.9681	0.7776	0.7533	0.7329	0.7718	0.7788	0.7704	0.7324	0.7418	0.7391	0.7400	0.7397	0.7323	0.7286	0.7268	**0.7011**

TABLE A.6: Cosine similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.7683	0.7000	0.6352	0.6284	0.6455	0.6312	0.6277	0.6354	0.6349	0.6347	0.6326	0.6357	0.6313	0.6313	0.6331	**0.6101**
F6	0.7555	0.7437	0.7115	0.6562	0.6485	0.6771	0.6692	0.6581	0.6405	0.6337	0.6661	0.6572	0.6516	0.6504	0.6458	**0.6176**
LM	0.7590	0.6204	0.6220	0.5977	0.5860	0.5974	0.6019	0.6115	0.5916	0.5849	0.5972	0.5913	0.5965	0.5912	0.5909	**0.5777**
NS	0.9517	0.7669	0.7476	0.7222	0.7696	0.7695	0.7635	0.7274	0.7248	0.7025	0.7007	0.7210	0.7411	0.7225	0.7155	**0.6872**

[1] St: Stemming option, Dl :Document length

[2] All document terms are used in the creation of document vectors.

TABLE A.7: Overlap similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.8039	0.7095	**0.6573**	0.693	0.7116	0.7486	0.7277	0.7221	0.7293	0.7219	0.7372	0.7642	0.74	0.7273	0.7207	0.7181
F6	0.8186	0.7211	0.7260	0.7082	0.6890	0.7123	0.7344	0.7153	0.7198	0.7338	0.7354	0.7251	0.726	0.7125	0.7017	**0.6794**
LM	0.7689	0.6738	**0.6589**	0.6867	0.6783	0.6998	0.7076	0.7125	0.7052	0.7105	0.7182	0.7107	0.7036	0.7088	0.6985	0.7033
NS	0.9361	0.7883	0.7682	0.7566	0.8309	0.8318	0.7829	0.7633	0.7385	0.7376	0.7461	0.7408	0.7282	0.7270	0.7103	**0.7001**

TABLE A.8: Dice similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.7841	0.6897	0.6184	0.6279	0.6627	0.6757	0.6630	0.6452	0.6422	0.6433	0.6347	0.6345	0.6398	0.6147	0.609	**0.5778**
F6	0.7494	0.7445	0.7300	0.6640	0.7002	0.7068	0.6960	0.6849	0.6757	0.6708	0.6703	0.6648	0.6497	0.6449	0.6449	**0.6139**
LM	0.7683	0.6246	0.6302	0.6245	0.6070	0.5916	0.6216	0.6051	0.5963	0.5953	0.5944	0.5907	0.5847	0.6105	0.6043	**0.5669**
NS	0.9616	0.7791	0.7566	0.7313	0.7711	0.7821	0.7701	0.7309	0.7420	0.7332	0.7365	0.7354	0.7319	0.7277	0.7200	**0.7002**

TABLE A.9: Hellinger similarity function experiments with training set

St / Dl[1]	10	20	30	40	50	60	70	80	90	100	120	140	160	180	200	All Terms[2]
F5	0.7642	0.7204	0.7068	0.7230	0.7846	0.7705	0.7783	0.7600	0.7779	0.7745	0.7849	0.7485	0.7647	0.7470	0.7297	**0.6852**
F6	0.7871	0.7665	0.7882	0.7819	0.8011	0.8069	0.8056	0.7873	0.7587	0.7761	0.7610	0.7517	0.7475	0.7581	0.7687	**0.7326**
LM	0.7838	0.6661	0.6731	0.7043	0.6820	0.7144	0.7207	0.7227	0.7447	0.7456	0.7156	0.6916	0.6679	0.6526	0.6429	**0.6207**
NS	0.9461	0.8211	0.7715	0.8531	0.8731	0.8578	0.8507	0.8584	0.8276	0.8224	0.7901	0.8041	0.8098	0.8286	0.8166	**0.7494**

Appendix E: Chronological Term Ranking - Parameter Selection Experiments

TABLE A.10: Additive parameter selection (C) experiments with training set

Additive Functions	**C = 0.0**	**C = 0.1**	**C = 0.2**	**C = 0.3**	**C = 0.4**	**C = 0.5**	**C = 0.6**	**C = 0.7**	**C = 0.8**	**C = 0.9**	**C = 1.0**
adp	0.5424	0.5188	0.5074	0.5014	0.4922	0.4927	0.4858	0.4858	**0.4762**	0.479	0.4833
adpl	0.5424	0.5241	0.5112	0.5049	0.5021	0.4897	0.4793	0.4776	**0.4771**	0.4923	0.4898
adpl2	0.5424	0.5284	0.5194	0.5144	0.5165	0.5055	0.4905	0.4825	0.4591	0.4701	**0.4559**
amp	0.5424	0.5398	0.5431	0.5395	**0.5393**	0.5394	0.5420	0.5427	0.5541	0.5563	0.5673
ampl	0.5424	0.5293	0.5100	**0.5079**	0.5096	0.5189	0.5292	0.5146	0.5146	0.5213	0.5081
ampl2	0.5424	0.5376	0.5298	0.5176	**0.5071**	0.5100	0.5152	0.5163	0.5166	0.5156	0.5176
ai	0.5424	0.5322	0.5223	0.5105	0.5044	0.4975	**0.4888**	0.5021	0.4937	0.4970	0.4943
ail	0.5424	0.5289	0.5117	0.5079	0.5019	0.5079	0.5066	0.4944	**0.4896**	0.4911	0.5096
ail2	0.5424	0.5293	0.5173	**0.5079**	0.5081	0.5093	0.5140	0.5199	0.5229	0.5174	0.5170

TABLE A.11:Multiplicative parameter selection (C) experiments with training set

Multiplicative Functions	**C = 0.0**	**C = 0.1**	**C = 0.2**	**C = 0.3**	**C = 0.4**	**C = 0.5**	**C = 0.6**	**C = 0.7**	**C = 0.8**	**C = 0.9**	**C = 1.0**
mdb	0.5424	0.5114	0.4811	**0.4790**	0.4798	0.4875	0.509	0.5062	0.5147	0.5125	0.5120
mdbl	0.5424	0.5232	0.4930	0.4789	**0.4767**	0.4966	0.5229	0.5175	0.5204	0.5191	0.5278
mdbl2	0.5424	0.5260	0.5114	0.4974	0.4791	**0.4581**	0.4735	0.4762	0.4908	0.4840	0.4908
mmb	0.5424	0.5312	0.5351	0.5327	0.529	**0.5168**	0.5289	0.5214	0.5214	0.5214	0.5214
mmbl	0.5424	0.5105	0.5202	**0.5040**	0.5191	0.5257	0.5233	0.5152	0.5205	0.5302	0.5371
mmbl2	0.5424	0.5297	0.5108	0.5229	0.5246	**0.5034**	0.5104	0.5195	0.5213	0.5215	0.5202
mds	0.5424	0.5103	**0.4888**	0.4975	0.5156	0.5174	0.5270	0.5245	0.5286	0.5401	0.5743
mdsl	0.5424	0.5229	**0.4933**	0.4983	0.5256	0.5203	0.5172	0.5452	0.5627	0.6220	0.6695
mdsl2	0.5424	0.5184	0.5086	0.4755	**0.4753**	0.5111	0.5119	0.5349	0.5749	0.6290	0.6904
mms	0.5424	0.5469	0.5293	0.5300	0.5286	0.5298	0.5324	**0.5214**	0.5237	0.5250	0.5235
mmsl	0.5424	**0.5105**	0.5164	0.5143	0.5148	0.5365	0.5524	0.5496	0.5662	0.5819	0.5944
mmsl2	0.5424	0.5297	0.5125	**0.5091**	0.5281	0.5243	0.5486	0.5623	0.5727	0.5719	0.5920

Appendix F: Statistical Tests of N-Pass Detection

TABLE A.12: Pair-wise statistical comparison results (p values) of additive functions

Function	Baseline	adp	adpl	adpl2	amp	ampl	ampl2	ai	ail
adp	0.065[1]	-	-	-	-	-	-	-	-
adpl	0.009[2]	0.037[2]	-	-	-	-	-	-	-
adpl2	0.091[1]	0.079[1]	0.001[2]	-	-	-	-	-	-
amp	0.208	0.121	0.025[2]	0.188	-	-	-	-	-
ampl	0.026[2]	0.491	0.144	0.245	0.039[2]	-	-	-	-
ampl2	0.015[2]	0.493	0.160	0.238	0.028[2]	0.434	-	-	-
ai	0.027[2]	0.178	0.436	0.085[1]	0.054[1]	0.191	0.186	-	-
ail	0.002[2]	0.002[2]	0.013[2]	0.001[2]	0.007[2]	0.010[2]	0.008[2]	0.007[2]	-
ail2	0.023[2]	0.481	0.155	0.223	0.034[2]	0.137	0.449	0.195	0.010[2]

TABLE A.13: Pair-wise statistical comparison results (p values) of multiplicative functions

Function	Baseline	mdb	mdbl	mdbl2	mmb	mmbl	mmbl2	mds	mdsl	mdsl2	mms	mmsl
mdb	0.031[2]	-	-	-	-	-	-	-	-	-	-	-
mdbl	0.004[2]	0.055[1]	-	-	-	-	-	-	-	-	-	-
mdbl2	0.064[1]	0.015[2]	0.030[2]	-	-	-	-	-	-	-	-	-
mmb	0.015[2]	0.134	0.011[2]	0.271	-	-	-	-	-	-	-	-
mmbl	0.001[2]	0.086[1]	0.274	0.020[2]	0.006[2]	-	-	-	-	-	-	-
mmbl2	0.004[2]	0.029[2]	0.464	0.009[2]	0.016[2]	0.176	-	-	-	-	-	-
mds	0.010[2]	0.308	0.011[2]	0.342	0.082[1]	0.031[2]	0.044[2]	-	-	-	-	-
mdsl	0.038[2]	0.118	0.039[2]	0.368	0.218	0.020[2]	0.010[2]	0.405	-	-	-	-
mdsl2	0.230	0.006[2]	0.008[2]	0.016[2]	0.384	0.003[2]	0.003[2]	0.064[1]	0.026[2]	-	-	-
mms	0.068[1]	0.213	0.028[2]	0.365	0.331	0.021[2]	0.032[2]	0.221	0.330	0.329	-	-
mmsl	0.064[1]	0.098[1]	0.005[2]	0.213	0.154	0.004[2]	0.011[2]	0.040[2]	0.175	0.462	0.171	-
mmsl2	0.037[2]	0.323	0.055[1]	0.376	0.152	0.010[2]	0.006[2]	0.482	0.432	0.095[1]	0.210	0.103

[1] Nearly significant

[2] Strongly significant