

NEW EVENT DETECTION AND TRACKING IN TURKISH

A THESIS

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

By

Süleyman Kardaş

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. Pınar Duygulu Şahin

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. H. Murat Karamüftüoğlu

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. Özgür Ulusoy

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

NEW EVENT DETECTION AND TRACKING IN TURKISH

Süleyman Kardaş
M.S. in Computer Engineering
Supervisors
Prof. Dr. Fazlı Can
Asst. Prof. Dr. Seyit Koçberber
May, 2009

The amount of information and the number of information resources on the Internet have been growing rapidly for over a decade. This is also true for on-line news and news providers. To overcome information overload news consumers prefer to track the topics that they are interested in. Topic detection and tracking (TDT) applications aim to organize the temporally ordered stories of a news stream according to the events. Two major problems in TDT are new event detection (NED) and topic tracking (TT). These problems respectively focus on finding the first stories of previously unseen new events and all subsequent stories on a certain topic defined by a small number of initial stories. In this thesis, the NED and TT problems are investigated in detail using the first large-scale test collection (BilCol2005) developed by Bilkent Information Retrieval Group. The collection contains 209,305 documents from the entire year of 2005 and involves several events in which eighty of them are annotated by humans. The experimental results show that a simple word truncation stemming method can statistically compete with a sophisticated stemming approach that pays attention to the morphological structure of the language. Our statistical findings illustrate that word stopping and the contents of the associated stopword list are important and removing the stopwords from content can significantly improve the system performance. We demonstrate that the confidence scores of two different similarity measures can be combined in a straightforward manner for improving the effectiveness.

Keywords: New Event Detection, Topic Detection and Tracking, Turkish.

ÖZET

TÜRKÇE'DE YENİ OLAY BELİRLEME VE İZLEME

Süleyman Kardaş
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticileri
Prof. Dr. Fazlı Can
Yrd. Doç. Dr. Seyit Koçberber
Mayıs, 2009

Son on yılda internet üzerindeki bilgi kaynakları sürekli artmaktadır ve buna bağlı olarak Web üzerinde bilginin büyüklüğü de hızla çoğalmaktadır. Bu, çevrimiçi haber ve haber kaynakları için de geçerlidir. Haber okuyucuları, daha çok ilgi duydukları konular hakkındaki haberleri okumayı tercih ederler. Yeni olay belirleme ve izleme (YOBİ) uygulamaları zaman sırasıyla gelmekte olan bir haber dizisinde yeni olaylara karşılık gelen ilk haberleri bulmayı ve bu ilk haberlerin tanımlanmış olduğu konular ile ilgili haberleri kümelemeyi hedefler. Literatürde YOBİ daha çok konu tanıma ve izleme (KTİ) adıyla bilinmektedir. Bu tezde, çok sayıda Türkçe Web haber kaynağından sağlanan haberler kullanılarak anında yeni olay belirleme ve izleme problemleri Türkçe için geliştirilmiş bu konuyla ilgili ilk büyük deney derlemi (BilCol2005) kullanılarak araştırılmıştır. Bu deney derlemi 2005 yılı 209,305 adet haberden oluşmuş olup 80 adet tekil konu ve ilgili hikayeleri ile birlikte insanlar tarafından etiketlenmiştir. Çalışmada Türkçe YOBİ işlemlerinde bazı benzerlik hesaplama katsayıları için her bir kelimenin ilk 5-6 harfin o kelimenin kökü olarak kullanmanın dilin morfolojik yapısını kullanan karmaşık bir kök bulma yaklaşım ile yarışabildiği gösterilmiştir. Aynı zamanda kelime durma listesi kullanımının sistem başarımını kayda değer düzeyde etkilediği ve iki farklı benzerlik hesaplama yönteminin sağladığı güvenlik skorlarının sade bir yaklaşımla birleştirilerek sistem etkinliğinin yükseltilebileceği gösterilmektedir.

Anahtar sözcükler: Yeni olay belirleme, yeni olay belirleme ve izleme, başarımların ölçülmesi, Türkçe.

Acknowledgement

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 would like to thank my second advisor Asst. Prof. Dr. Seyit Koçberber for his support and help throughout the work.

I would like to thank all my Information Retrieval Group team members, specifically Özgür Bağlıoğlu, Hüseyin Çağdaş Öcalan, and Erkan Uyar for their helps and supports.

I would like to thank TÜBİTAK for its partial financial support of my thesis work under the grant number 106E014 ("Instant new event detection and tracking and retrospective event clustering in Turkish news using Web resources"). I would also like to thank Computer Engineering Department of Bilkent University for the same reason.

I would like to thank National Research Institute of Electronics and Cryptology (TÜBİTAK UEKAE) for giving this great opportunity.

to my Mother, Father and little sweet niece (Esma Hilal)

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Research Contributions	4
1.3	Overview of the Thesis	5
2	Related Work	6
2.1	New Event Detection Approaches	6
2.1.1	The CMU Approach	7
2.1.2	The UMass Approach	7
2.1.3	The Dragon Approach	9
2.1.4	The UPenn Approach	10
2.1.5	The BBN Technologies Approach	11
2.1.6	The BFC Approach	11
2.1.7	The MMS Approach	12
2.1.8	The LTY Approach	12

2.1.9	The ZZW Approach	13
2.1.10	Kurt's Thesis: On-Line New Event Detection in Turkish	13
2.1.11	Vural's Thesis: On-line New Event Detection	14
2.1.12	Summary of New Event Detection Approaches	14
2.2	Event Tracking Approaches	15
2.2.1	The CMU Approach	15
2.2.2	The UMass Approach	17
2.2.3	The Dragon Approach	18
2.2.4	The UPenn Approach	19
2.2.5	The BBN Approach	19
2.2.6	The MMS Approach	20
2.2.7	The ZZHFL Approach	20
2.2.8	Profile-based Event Tracking	21
2.2.9	BuzzTrack: Topic Detection and Tracking in Email	22
3	NED and TT Methods Used in the Thesis	23
3.1	New Event Detection	23
3.1.1	Document Preprocessing	24
3.1.2	Document Representation	26
3.1.3	Document Comparison	27
3.1.4	Time Window	29

3.1.5	Novelty Decision	30
3.1.6	New Event Detection Algorithm	31
3.2	Topic Tracking	31
3.2.1	Adaptive Tracking Approach	32
3.2.2	Combining Cosine and CC Similarity Scores for Tracking .	32
3.2.3	Topic Tracking Algorithm	33
4	Experimental Evaluation	34
4.1	Data Set	34
4.2	Evaluation Methodology	35
4.3	New Event Detection Results	38
4.3.1	Vector Length, Stemmer, and Window Size Tuning	38
4.3.2	Effects of the Stopword list	40
4.3.3	Effect of Window Size	41
4.3.4	Intuition Behind our Combination Methods: Decision Con- sistency of CC and Cosine Similarity measures	43
4.3.5	Training Results	45
4.3.6	Evaluation Results	46
4.4	Topic Tracking Results	47
4.4.1	Determining Vector Lengths and Stemmer	47
4.4.2	Training Results	47

<i>CONTENTS</i>	xi
4.4.3 Test Results	49
4.5 Chapter Summary	50
5 Further Experiments	52
5.1 Influence of other Similarity Measures on NED	52
5.2 Effect of Time Penalty Function	57
5.3 Effect of Using Named Entities	58
5.4 Chapter Summary	59
6 NED and TT in Bilkent News Portal	60
6.1 News Resources	60
6.2 Changes in NED and TT Methods for Bilkent News Portal Imple- mentation	62
6.2.1 News Categorization Before NED and TT	62
6.2.2 Limiting Number of Topics for an Incoming Story	62
6.2.3 Using Named Entities During TT	63
7 Conclusions	64
A Stopword Lists	74
B Annotated Events	76

List of Figures

1.1	Bilkent News Portal.	4
2.1	Single-Pass clustering algorithm.	8
2.2	UMass approach for on-line new event detection algorithm.	9
2.3	Dragon approach for new event detection system.	9
2.4	UPenn Approach to new event detection.	10
2.5	BBN approach for new event detection algorithm.	11
3.1	Sliding time-window approach in TDT.	30
3.2	The NED algorithm used in the thesis.	31
3.3	The static TT algorithm used in the thesis.	33
3.4	The adaptive TT algorithm used in the thesis.	33
4.1	The distribution of BilCol2005 topic stories among the days of 2005.	35
4.2	Min. C_{Det} with different time-window sizes for CC and cosine measures.	42

4.3	Confidence scores of first (left figure) and tracking (right figure) stories. In CC, the confidences score 1 corresponds to a CC value 1 or higher.	43
4.4	Consistency of CC and cosine decisions in NED during training for 50 events.	44
4.5	NED training performance of CC and cosine measures.	45
4.6	Static TT training Min. C_{Det} of CC and cosine similarity ($N_t = 1$).	48
6.1	Daily number of news for each news resources.	61

List of Tables

2.1	Summary of new event detection approaches	14
4.1	Distributions of stories among training and test sets	38
4.2	Min. C_{Det} values for CC and cosine measures for ned with various vector lengths and stemmers	39
4.3	Effects of stemmer on effectiveness with CC and cosine measures .	40
4.4	Effects of stopword list size on effectiveness with CC and cosine measures	40
4.5	NED training results	46
4.6	NED test results	46
4.7	Static TT training results	48
4.8	Adaptive TT training results	49
4.9	Static TT test results	49
4.10	Adaptive TT test results	50
5.1	NED standalone use of similarity measures training and test per- formance	54

5.2	NED and-combination training performance Min. C_{Det} (upper no.) and test C_{Det} (lower no)	55
5.3	NED or-combination training performance Min. C_{Det} (upper no.) and test C_{Det} (lower no.)	55
A.1	Stopword list (10 words)	74
A.2	Stopword list (147 words) [20]	74
A.3	Stopword list (217 words)	75
B.1	Summary information for annotated events	76
B.2	(Cont.) Summary information for annotated events	77

Chapter 1

Introduction

1.1 Motivations

Information explosion has new dimensions with the advances in information technologies. For example, the number of news resources on the web has exponentially increased in the last decade. Multi-resource news portals, a relatively new technology, receive and gather news from several web news providers. More advanced versions of these portals aim to make news stories more accessible by providing event-oriented groupings of news stories. Topic detection and tracking, (TDT), applications aim to organize the temporally ordered stories of a news stream as they arrive. Such event-based organizations facilitate an abstraction and aim to prevent overwhelming of news-consumers that can be caused by too many unconnected stories (in the rest of the thesis the words news, story, and document are used interchangeably). Services on current events are a popular activity on the web. For instance, recently Wikipedia, which is a non-profit organization, has introduced a current events portal and it has approximately 3 million articles [4]. Commercial news portal examples with such services include Google News [2] and NewsIsFree [3] and research-oriented examples include NewsInEssence [51] and NewsBlaster [43].

Information retrieval, IR, and TDT have some similarities but they involve

two different types of information organizations. IR deals with subject-based information organizations and needs. For example, the subject ‘flowers that grow in shade’ [8] involves no event. What we know about a subject may change with time; however, stories related to a subject (e.g., such flowers) are always about the subject. The temporal nature of events is the most prominent difference between subject-based and event-based information organization approaches. In event-based information organization, the notion of relevance can shift with time. [7] gives the example of ‘Oklahoma City bombing’, the event was first regarded as ‘Middle East terrorism come to the US’ and a few days later this was no longer appeared in the news and the topic is associated with ‘right-wing militia groups in the US’. As a result, the topics are seen to evolve over time and they change.

TDT defines an event as something that happens at a given ‘place and time’. It need not involve the human participation of human actors. It is a concept that is studied in detailed by philosophers [58]. According to the philosophers, in a metaphysical sense, events take place when there is a conflict between physical objects [48]. In TDT studies, topic is defined as ‘a seminal event or activity with all directly related events and activities’. In this context, an activity is defined as ‘a connected set of actions that have a common focus or purpose’ [48].

The most influential activity in this field is the TDT research initiative. It was pursued under the DARPA TIDES (Translingual Information Detection, Extraction, and Summarization) program with various workshops [30]. The TDT research initiative tasks are described as follows [8].

1. *New Event Detection* (NED): aims to instantly identify the occurrence of the first story for a new event.
2. *Topic Tracking* (TT): aims to find subsequent stories on a topic in a news stream using one or more sample stories. TT involves supervision due to training provided by the sample stories.
3. *Topic Detection* (TD): aims to build story clusters as news stories arrive based on topics they discuss using the information provided by automatically detected first stories of new events. This task is also known as ‘cluster

detection’. TD is similar to TT, but completely unsupervised: there is no sample story provided by humans.

4. *Story Segmentation* (SS): aims to divide transcription of a news broadcast into individual stories.
5. *Story Link Detection* (SLD): aims to detect if two randomly selected stories discuss the same news topic.

The most heavily studied subjects of TDT are the first three tasks. As indicated above TD and TT are similar tasks: in TT sample stories are provided by humans; on the other hand, in TD tracking is based on first stories detected by the system [8]. It is expected that a method that performs well in TT would also be an effective TD method, provided that the first stories are properly selected. During NED, some of the tracking stories of some 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. This causes a circular problem: a poor TT system would miss tracking news of existing events and such news could be falsely identified as first story of false new events. As a result, this would further decrease the tracking performance, of course, at the same time would also decrease the first story detection performance. In other words, there is a direct relationship between performances of NED and TT [7].

There has been enormous study on TDT in English, however; there are almost no study on TDT in Turkish. For this reason, this thesis focuses on mainly researching the TDT problems on Turkish with a large scale experimental corpus. The main purpose of this thesis is to provide an automatic environment for constructing a Turkish News Portal. Findings of these thesis and the other related thesis [13, 46, 59] are used for the implementation of Bilkent News Portal (<http://newsportal.bilkent.edu.tr /PortalTest/>) [19]. The main page of the portal is depicted in Figure 1.1. It can be seen that there is a tabbed panel labeled as ‘Recent & Past Event’ on the right corner of the portal. The ‘Recent Event’ pane tab illustrates the all new event currently captured by the system but the

‘Past Event’ shows the events with high number of tracked stories were captured before. The technical features of the news portal is discussed in [46].



Figure 1.1: Bilkent News Portal.

1.2 Research Contributions

In this thesis we

- Use a large scale Turkish corpus in the experimental evaluation of NED and TT tasks under several different conditions in terms of the similarity measures, number of features used in document description vectors, and word stemmers.

- Introduce a new similarity measure for TDT and compare it in detail with the traditional vector space-based cosine approach.
- Combine the new similarity measure results and the cosine results to obtain a better performance.
- Report the results of our investigation with several other similarity measures and provide the results of our experiments on additional issues such as the use of named entities and the reflection of the age of the stories to the calculation of the similarity values.
- Supply some practical recommendations for the implementation of TDT systems in the Turkish language.

1.3 Overview of the Thesis

Chapter 2 provides a review of some related studies on TDT. Chapter 3 introduces our algorithm for NED and TT. Chapter 4 describes our test collection and methodology for comparison and presents experimental results covering NED and TT. In Chapter 5, we present the results of some addition experiments that we tried. Chapter 6 provides reflections from the Bilkent News Portal whose implementation is based on the findings of this thesis and the works of the other members of the Bilkent Information Retrieval Group [13, 46, 59]. Finally, in the last chapter, we conclude the work and provide some future research pointers.

Chapter 2

Related Work

Topic Detection and Tracking (TDT) was started in 1997 with a Pilot study and has continued with open evaluations in TDT 1998-2004 . The main purpose of the TDT is to develop core technologies in order for a news understanding systems [30]. Therefore, a pilot corpus was collected and annotated, evaluation procedures were developed and novel research was conducted with a highly cooperative joint effort with Carnegie Mellon University, DARPA (US Department of Defense Advanced Research Project Agency), Dragon Systems, and the University of Massachusetts at Amherst [62]. It was later pursued under the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program. The first technical approaches and TDT results were published in 1998. With TIDES, a forum is introduced in order to discuss applications and techniques for on-line new event detection and tracking problem that has not been studied prior to the TDT research efforts[30, 48]. In the next two sections, some approaches to NED and TT problems are introduced.

2.1 New Event Detection Approaches

In TDT, small amount of deferral period (less than 10 news stories) before a decision is considered as in immediate mode. We used immediate mode in our

on-line new event detection (ONED). Therefore, in this section, we provide only the approaches with immediate mode.

2.1.1 The CMU Approach

The CMU (Carnegie Mellon University) uses conventional vector space model to represent a story and cluster in their NED approach. The most common document representation, which is a vector of weighted terms whose dimensions are unique terms from dataset and whose elements are the term weights, is used in their new event detection. Term weighting occurs along with rules where high frequency terms that appear in many stories are weighted lower than the terms that are important in a particular story but not in the rest of the collection.

They utilized cosine similarity function to compare a story with a cluster. Single-pass or incremental clustering is used to create clusters online and the algorithm [60] is described in Figure 2.1. In the algorithm, stories are processed sequentially one at a time. Each incoming story is compared with existing clusters and it is merged with the most similar clusters if the similarity is above pre-determined threshold; otherwise, the story is treated as the seed of new cluster. Moreover, CMU is utilized a linear decaying function in order to decrease the influence of the clusters in time on the decision. They also use sliding time-window to limit prior context to pre-defined m preceding stories.

2.1.2 The UMass Approach

In the UMass approach, several text representation issues are explored in the context of event tracking, where a classifier for a topic, consist of one or more sample stories. The classifier is used to discriminate related subsequent stories from stream. Several different topic representation techniques and feature selection methods are discussed. Query formulation involved a three steps process:

1. The documents are processed sequentially.
2. The representation for the first document becomes the cluster representative of the first cluster.
3. Each subsequent document is matched against all cluster representatives existing at its processing time.
4. A given document is assigned to one cluster (or more if is allowed) according to some similarity measure.
5. When a document is assigned to a cluster, the representative for that cluster is recomputed.
6. If a document fails a certain similarity test, it becomes the cluster representative of a new cluster

Figure 2.1: Single-Pass clustering algorithm.

1. Term selection
2. Weight Assignment
3. Threshold estimation

They also point out that performance of the system can be improved by using named entities and temporal information of stories. In the first story detection system, a modified version of the general-purpose single-pass clustering method is used [12, 60]. The algorithm is depicted in Figure 2.2 that processes each incoming story, whose content is represented as a query in the assumption of discussing some event, on the stream sequentially [50].

Moreover, they explore 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. The outcomes suggest that the online single-link strategy extended with a model for domain properties is faster and more effective than other cluster approaches. In the experiments, they utilize the INQUERY [16] information retrieval system whose performance is tuned for TDT with some intuitive parameters based on experimental observations.

1. Use feature extraction and selection techniques to build a query representation to define the story's content.
2. Determine the query's initial threshold by evaluating the new document with the query
3. Compare the new story against previous queries in memory
4. If the document does not trigger any previous query by exceeding its threshold, flag the story as containing new event
5. If the document triggers an existing query, flag the story as not containing new event.
6. (Optional) Add the story to the agglomeration list of queries it triggered.
7. (Optional) Rebuild existing queries using the story.
8. Add new query to memory.

Figure 2.2: UMass approach for on-line new event detection algorithm.

2.1.3 The Dragon Approach

In the Dragon detection system, a story and clusters representations are based on a language model that use unigram word frequency distribution. The cluster unigram distribution was smoothed by means of absolute discounting [14] with a common unigram distribution built from background distribution. The detection algorithm is rooted in a simple k-means clustering strategy and the algorithms are depicted in Figure 2.3.

1. At the moment an incoming story is received, assume that there are k story clusters, each cluster characterized by a set of statistics. The distance between each of these clusters and the story is computed.
2. The story is inserted into the closest cluster, unless its distance to this cluster exceeds a threshold, in which case a new cluster is created.
3. Before accepting another incoming story, the cluster assignments of all uncommitted stories (stories for which the incoming story is within their deferral period) are reevaluated. Stories may move, seed new clusters, or remain in their assigned cluster. This step is repeated as often as desired.

Figure 2.3: Dragon approach for new event detection system.

They use a new distance measure, named Kullback-Leibler [5, 64], to compare

a given story and existing cluster in the detection system. They also use decaying concept in their detection in order to cause clusters to have a limited existence in time. This provides system tunable by adjusting the decay parameter.

2.1.4 The UPenn Approach

The UPenn approach utilized the idf-weighted cosine coefficient described in [55] in comparison of a story and a cluster. The stories and clusters are represented as vectors in n -dimensional space, whose dimension (n) is the number of unique terms in the database and whose coefficients the term frequencies in the story. They ignored high frequent words in the database for the reason that the resulting vector is enormously sparse.

Their clustering technique is based on single linkage clustering. In this clustering methodology, all stories initially begin in their own singleton clusters. Two clusters are merged if the similarity value between any story of the first cluster and any of the second clusters is above a pre-determined threshold. The purpose of the detection task is to identify stories pertaining to the same event without aid of either positive or negative feedback. The algorithms is shown in Figure 2.4.

1. A deferral period is defined to be the number of files (including a number of stories) the system is allowed before it relates an event with the stories of that file. Inverted index is then created.
2. For each story, term frequencies are sorted and then n most frequent ones are used to create document vector.
3. All stories are compared to the preceding ones, including stories from a previous deferral period.
4. If the similarity is exceeds a certain threshold, their clusters are merged.
5. If a story cannot be combined with any other existing cluster, it turns into a new cluster (new event)

Figure 2.4: UPenn Approach to new event detection.

2.1.5 The BBN Technologies Approach

In the BBN system, an array of effective probabilistic models is introduced so as to measure the similarity between a story and a topic. They have invented four different similarity measures, which are described in detail in [37]. They also propose a useful solution to the difficulty of score normalization method to improve scores.

In the detection system, a customized incremental k-means algorithm is utilized to cluster stories where no story can be on more than one topic cluster. In this clustering method, number of the clusters is not specified in advance. It is determined in the processing last incoming story. The detection algorithm is shown in Figure 2.5.

1. Use incremental clustering algorithm is depicted in Figure 2.1 to process stories up to the end of the current modifiable window.
2. Compare each story in the window with old cluster to determine whether each story should be merged with that cluster or used as a seed for new cluster.
3. Modify all the clusters at once according to the new assignments.
4. Iterate steps (2)-(3) until the clustering does not change.
5. Look at next few stories and go to 1.

Figure 2.5: BBN approach for new event detection algorithm.

2.1.6 The BFC Approach

Brants, Chen, and Farahat (BFC) introduce a new method for performing NED in one or multiple news streams [15]. They extend a 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. Their method is based on incremental TF-IDF model.

Experimentally, they tested replacing cosine distance by Hellinger distance,

source specific tf-idf model, and source specific similarity normalization on the system effectiveness. They observed that each additional element only yields a slight improvement in topic-weighted normalized minimum cost but taken together it provides about 18% higher performance than that of their baseline approach. Moreover, they also tested the use of vocabulary in the look-ahead data (also called the deferral period that can be 1, 10, 100) for the tf-idf model and the use of the time information, however; they found that these two techniques did not improve the system effectiveness.

2.1.7 The MMS Approach

Makkonen, Myka and Salmenkivi (MMS) suggest a method that incorporates simple semantics into TDT by splitting term space into some meaningful semantic groups that include proper names, locations, and temporal expressions [41]. Such groups can provide important clues about news and ordinary terms. This grouping makes it possible to compare two documents class-wise, and assigning each class a dedicated similarity measure, which can utilize external ontology. They build a geographical and temporal ontology that use of which relies on extensive use of natural language processing techniques. They use cosine similarity in comparison of two documents. Moreover, they include time expressions in their feature groups to improve detection and tracking performances. They show the positive impact of the use of named entities in TDT.

2.1.8 The LTY Approach

Luo, Tang, and Yu (LTY) study a practical new event detection system using IBM's Stream Processing Core middleware [40]. They consider both effectiveness and efficiency of such a system in practical settings that can adapt itself according to availability of various system resources such as CPU time and memory.

They evaluated the effectiveness of their techniques on the standard TDT5

benchmark. In their experiments, they use, as a baseline, a variant of the state-of-the-art Okapi formula to compute both term weights and the similarity values of document pairs. They also used sliding-window concept in the detection. Luo et al. state that their work is the first implementation of an online new event detection application in a large-scale stream processing system.

2.1.9 The ZZW Approach

Zhang et al. (ZZW) propose a new method to speed up new event detection by using a dynamically indexing tree [67]. They use incremental TF-IDF model for term weight calculation and use Hellinger distance for comparison of two stories in the baseline. They also propose two term reweighting approaches using term type and statistical distribution distance. In their first approach, they propose to adjust term weights dynamically based on previous story clusters and for the second approach, they suggest to employ statistics on training data to learn the named entity reweighting model for each class of stories. TDT2 and TDT3 datasets are used in the experiments. Experimentally, they observed their approaches significantly improve both efficiency and effectiveness, compared to the baseline.

2.1.10 Kurt's Thesis: On-Line New Event Detection in Turkish

There is little research on TDT in the Turkish language. This is partly due to the fact that there is no standard test collection for Turkish. To the best of our knowledge [36] has conducted the only TDT study for Turkish other than ours using 46,530 stories belonging to the first three months of 2001 from four news resources provided by the Reuters news feed. His test collection contains 15 annotated events with about 88 stories per event (min. 11, max. 358 stories).

For the new event detection algorithm, Kurt proposed a method that is a

combination of the single-pass and k-NN clustering algorithms and uses the time-window concept. In comparing a story and a topic, cosine similarity measure is used. Moreover, in order to decrease the influence of stories in time, he utilizes time penalty function on decision score.

2.1.11 Vural's Thesis: On-line New Event Detection

Vural, in his thesis, employs the concepts of the cover coefficient-based clustering (C3M) methodology [17, 22] for on-line new event detection and event clustering. The main purpose of his study is to use modified cluster seed power concept in detecting the first stories. He also uses sliding window in detection in order to prevent producing oversized event cluster and to give all documents to be the seed of a new event [61].

2.1.12 Summary of New Event Detection Approaches

The new event detection approaches that are described in the previous section are summarized in the following Table 2.1 where N/A means no information available. VSM refers to vector space model and SSM is source specific model. SM also refers to Semantic Model.

Table 2.1: Summary of new event detection approaches

Institution	Weighting	Story	Clustering	Similarity	Sliding Time window
CMU	TF-IDF	VSM	SinglePass	Cosine	used
Umass	TF-IDF	VSM	SinglePass	Inquery	used
Dragon	N/A	N/A	k-means	Distance Measure	used
UPenn	TF-IDF	VSM	Nearest Neighbor	Cosine	N/A
BBN	Probabilistic	VSM	Incremental k-means	Probability	N/A
BFC	TF-IDF	SSM	SinglePass	Hellinger	N/A
MMS	TF-IDF	SM	SinglePass	Cosine	used
LTY	TF-IDF	VSM	SinglePass	Okapi	used
ZZW	TF-IDF	VSM	Indexing Tree	Hellinger	N/A
Kurt	TF-IDF	VSM	SinglePass & kNN	Cosine	used
Vural	Probabilistic	C3M	SinglePass	Probability	used

2.2 Event Tracking Approaches

According to TDT, the tracking task, which is fundamentally similar to the standard routing and filtering tasks of Information Retrieval (IR), is defined to be the associating incoming stories with events known to the system. An event is described *known* by its association with stories that discuss the event. Therefore, each target event consists of a list of stories that discuss it [44]. In the tracking task, a set of on-topic stories are given and a portion of the evaluation corpus are also given to train models on. The tracking system must train and test on each topic independently.

A most important task parameter is the number of stories used to define the target event, N_t . Specifically, training set for a particular event and a certain value of N_t story will be all of the stories up to and including N_t^{th} story that discuss it. The TDT limited the number N_t to be 1, 2, 4, 8 and 16 [30]. In the literature, N_t is mostly used as 1.

2.2.1 The CMU Approach

In the CMU tracking approaches, Rocchio, K-Nearest-Neighbor (KNN), and Decision Trees (DT) approaches are utilized [23].

Rocchio, which is a common approach in information retrieval, uses a vector to represents each class and story. It calculates their similarity via the cosine coefficient values of these two vectors, which consist of the weights of terms (words or phrases), and obtains a binary decision by thresholding on this value. They use a common TF.IDF weighting scheme describes in [54].

The vector representation for a topic is called centroid. It is constructed from a set of (R) positive training story (u) and a set of (N) negative training stories (v) of that class and T is training set. The Set $\bar{R}$ consists of n top ranking stories retrieved from negative stories when using the centroid of the positive stories as the query. The centroid vector is represented as:

$$\vec{c}(\gamma, n) = \frac{\sum_{u \in R} \vec{u}}{|R|} + \gamma \frac{\sum_{v \in \bar{R}} \vec{v}}{n}$$

where n is the size of local zone and γ is weight of the negative centroid. The scoring function for a test story $\vec{x}$ with respect to the class $\vec{c}$ is defined in the following function.

$$\cos(\vec{x}, \vec{c})$$

In the tracking, they adopted conventional M-ary kNN approaches, which has been applied commonly in text categorization, specifically by Yang [65]. The kNN uses training stories ‘local’ to each test story to classify it. Since each topic should be tracked independently, they trained a kNN classifier per topic separately. Fundamentally, their system compares each incoming story with training stories and selects k-nearest neighbors based on the cosine similarity. The similarity scores on the input story is computed by summing the similarity scores for the positive and the negative stories respectively in the k-neighborhood and the difference is the score. If this score exceeds a certain threshold, then the incoming story is assumed to be the tracking news of the target event.

The number of positive stories per topic is extremely small whereas the number of the negative stories is high enough. This makes system difficult to achieve fewer misses without sacrificing more false alarms. Therefore, they use two modified version of the kNN in their system. First method is to decrease the influence of the negative stories by sampling a small portion in the k-neighborhood and ignoring the remaining stories. Second solution to this problem is taking average the similarity scores of two subsets. The two variants have different performance characteristics; one tends to produce high-precision results while the other yields better recall benefits.

The CMU also uses a decision tree learning algorithm (DTree), which is generally implemented in text categorization. The DTree employs the words and their some meta-feature such as multiple occurrences of word, highly allocated bigrams

and *et* as features. A decision tree was constructed for each topic using the recursive partitioning algorithm with maximum information gain splitting rule. In the splitting rule, the N top-ranked features from the collection are picked up and then the collection of stories is split according to the decision node whether or not they contain at least M of those N features [23].

2.2.2 The UMass Approach

In the tracking task, they used the training data with positive and negative stories to create the event being tracked. Each event is represented the same query representation for event detection. They also utilized the training data to derive a threshold for comparison with that query. This query was applied to all successive stories and then if any of them matches, they are assumed to be tracked. Moreover, they employ normalization process over similarity values because their similarity function produces different distribution of values for different topics. Two different tracking approaches are tested.

First one is static approaches where query formulations where query terms and thresholds are held constant over time. Static query formulation served as their baseline system during the training and development stages. Queries are formulated for each topic using n most frequent non-stop words from the on-topic stories in the training data. The words are given weights using assignment based on *tf* [cite]. Furthermore, they tested static queries with multiword features and tested two weight [49] - learning algorithms: Dynamic Feedback Optimization (DFO) [56] and Exponentiated Gradient Descent (EGD) [38].

Second approach is adaptive one where stories are assumed to on-topic are used to re-formulate the query over time. The queries are originally formulated using static approach and then reformulated on-line with features from new stories in the stream. Therefore, both on-topic and off-topic documents are saved and used to reformulate a query. In this approach, two thresholds are used. One threshold is estimated for detection decision and another threshold is estimated for agglomeration decision. If a story is above second threshold, it is employed

to reformulate the query and determine new threshold. They found that reformulation queries with 10 terms worked better than 20 and 50 terms. In addition, they experimentally show that adaptive approaches works better than static one if the number of training story is high ($N_t \geq 4$)

2.2.3 The Dragon Approach

The Dragon tracking approaches uses two statistical models; one is a topic model built from the supplied topic training stories and another is a discriminator model built from available training corpus. The tracking score of a test story is described as the log ratio of the probability that the topic model constructed the story is opposed to the probability that story was generated by the discriminator. This tracking score can be interpreted as the distance measure between an incoming news story and a topic model that consists of training story collection.

In the tracking system, a topic is modeled as a simple unigram distribution, $p(w)$. Then the probability that a test story was generated by $p(w)$ is just the product of the probabilities of each word being drawn from this distribution :

$$P_T = \prod_{w \in T} p(w)$$

Topic model were built from a list of the words, in which the common stopwords are removed, in the training stories. Since topic model contains small number of words, typically fewer than 1000 words total, they used a unigram smoothing make use of a technique called target [63]. In targeting smoothing, a large number of background unigram distributions are taken to find the mixture of these that best approximate the topic model. Specifically, a topic model is consisting of topic training stories and a set of background model. On the other hand, the discriminator model is also unigram distribution built from a large amount of training corpus. Furthermore, the performance unsupervised adaptation techniques are also explored. In TDT 2000 experiments, their adaptation method has improved the performance on some larger topics at the expense of smaller topics.

2.2.4 The UPenn Approach

The UPenn approach utilizes the same text representation used in the event detection. They also used the same *TF.IDF* weighting scheme, comparison methodology and feature selection that are described in section 2.1.4. They investigated four methods of producing lists of features sorted by their effectiveness in discriminating a topic and this made the number of the features vary for the topic vectors. Therefore, training corpus is employed to optimize the number of n most frequent terms for each case of N_t . They used only unsupervised static approach in their tracking task. They have tried a number of approaches to optimize *TF.IDF* weighted cosine coefficient for tracking task. Experimentally, it is found out straightforward feature selection with no normalization of topic scores achieved best.

2.2.5 The BBN Approach

The BBN's tracking system is built from the same the core technologies in the event detection. Their system is rooted in formulating a combination of three models: Topic Spotting (TS), Information Retrieval (IR) and Relevance Feedback (RF). In the first two approaches, a probabilistic approach to word occurrence distribution is used. In the TS model, the probability that words in the test story were generated by the training stories is computed. In the IR model, the probability that the training stories were generated by the words of the test story is computed. On the other hand, in the RF model, the most frequent terms occurred in the training stories are used in computation of similarity.

Moreover, time penalty is incorporated in the probability that an incoming story is on-topic in order to decreases as time increases. Therefore, most topics are alive in only a limited span of time.

Furthermore, the BBN uses adaptation in their tracking. They add the topic any stories whose scores exceed a certain threshold (θ_{adapt}), which is higher than on-topic threshold (θ_{on}). The adaptation is just a matter of re-estimating a

unigram distribution for the topic cluster.

2.2.6 The MMS Approach

Makkenon et al.'s (MMS) tracking system is built from the same core technologies in the event detection [41]. They do not update document representation. In the MMS tracking system, they use a straight-forward topic tracking algorithm. First of all, they construct the event vector for the given topic from the sample training stories. Then they go through the incoming documents one-by-one, build the event vector and compare it class-wise against the topic vector. Each class-wise similarity is multiplied with a different pre-calculated weight. If sum of all these multiplied class-wise similarities exceeds a pre-determined threshold, the story is assumed to be track of the target topic.

Unlike NED, in the experiments, the results indicated that the semantical augmentation degrades the performance of the tracking system. However, they suspect that this is at least partially due to the inadequate spatial and temporal similarity functions .

2.2.7 The ZZHFL Approach

In Zhang et al.'s (ZZHFL) topic tracking approach, a new method is proposed to build the keywords dependency profile (KDP) of each story and track topic based on similarity between the profile of topic and story [68]. The keywords are extracted from a story by a document summarization technology (MEAD). The goal of the MEAD is to find the indicative sentence of the story. The keywords of the selected sentences are the candidate words that are used to produce KDP. They construct a graph that consists of edge and nodes. Node represents the candidate words and edges represent co-occurrence of any two words (nodes) in the same sentence. The weights of nodes and edges calculated and these weights depend on two factors.

1. The weight of a keyword is high if it strongly depends on other ‘important’ keywords and the initial ‘importance’ of these keywords is computed by their tf-idf values.
2. A keyword is important if it mostly co-occurs with other keywords.

KDP keywords, edges, weights are used to construct a story model. The topic model is built with one or multiple sample stories. They used their own similarity measure with combining cosine similarity in comparison of a story model with a topic model. They used the CMU tracking system as their baseline system, which won the first place in TDT5 evaluation. The CMU approach used an improved variant of Rocchio algorithm to build topic model [24]. They tested their approaches on the mandarin resource of TDT4 and TDT5. The experimental results indicate that the topic tracking system based on KDP improves the performance by 13.25 % on training dataset and 7.49% on testing dataset comparing to baseline.

2.2.8 Profile-based Event Tracking

In this research, a profile-based event tracking method is proposed [39]. They attempt to build an event profile from the given on-topic stories by robust information retrieval technologies. A feature selection metric and a recognized event clause are utilized to determine most key semantic elements of the target event. In construction of an event profile, named entities such as location, date, person name and organization, are the most important elements. They use cosine similarity measure for comparison of a story with an event profile.

They tested their approaches on the TDT2 mandarin corpus and the results indicate that this profile-based event tracking method is promising. The optimal normalized detection cost drops from 0.0656 to 0.0390.

2.2.9 BuzzTrack: Topic Detection and Tracking in Email

In this research, topic tracking is utilized to produce an email client extension (BuzzTrack) that is to help users cope with email overload [27]. In this extension, the authors propose a clustering algorithm that creates the topic based grouping, and a heuristic for labeling the resulting clusters to summarize their content. For the email representation, vector space model is used. Each term, in the vector, is weighted using a standard TF.IDF formula. They used weighted overlap similarity function in comparison of an incoming email and existed topic cluster.

They create a collection corpus, which consist of 1586 emails with 537 topics for development set and 817 emails with 269 for test set, in order to evaluate their approaches. The experimental topic tracking results are comparable in performance to that of current work on TDT in news article[27].

Chapter 3

NED and TT Methods Used in the Thesis

3.1 New Event Detection

New event detection (NED), which is sub-task of TDT initiative, requires identifying new stories that discuss an event that has not been previously detected. NED works without a predefined query. The algorithm behind NED commonly searches keywords in a news story and compares the story with earlier stories. Our new event detection algorithm is based on a single-pass clustering algorithm presented in [9]. Our NED does not cluster stories in any way. The content of each incoming story is represented as a query consists of keywords. When a story is processed on the algorithm, all the previous stories are compared with this query and in this comparison a similarity between this query and other query is produced. If the computed maximum similarity is below a predetermined threshold, the target story is assumed to be a ‘*NEW*’ event. Otherwise, it is a continuation of story and so it is labeled as ‘*OLD*’. In the remainder of this section, the processes in our NED algorithm are explained.

3.1.1 Document Preprocessing

Like in text mining area, document preprocessing stage is the first step of our implementation. In this stage, each incoming story is tokenized into words. Then each word is cleaned of digits, punctuation marks and the case of letter. After that stopwords that carry no information are eliminated. Finally, the remaining words are stemmed with objective of removing suffixes from word root so as to increase the system performance. To sum up, this stage consists of following steps:

1. Tokenizing
2. Cleaning
3. Stopword elimination
4. Stemming

In the next sections, stopwords elimination and stemming processes are discussed in detail.

StopWord Elimination

In IR, a stopwords list contains frequent words that are ineffective in distinguishing documents from each other. In IR, elimination of such words decreases index size, increases efficiency, and can improve effectiveness [26]. In [34], they show positive influence of category-based word stopping on NED. We investigate the effects of word stopping by using four stopwords lists with different sizes. As a baseline, we first experiment with a stopwords list that contains no words. It is the no stopwords list case. This is followed by a stopwords list that contains ten most frequent words as determined in [21]. These words in decreasing frequency order are ‘ve,’ ‘bir,’ ‘bu,’ ‘da,’ ‘de,’ ‘için,’ ‘ile,’ ‘olarak,’ ‘çok,’ and ‘daha’ (their meanings in the same order are ‘and,’ ‘a/an/one,’ ‘this,’ ‘too,’ ‘too,’ ‘for,’ ‘with,’ ‘time,’ ‘very,’ and ‘more’). We also experiment with the semi-automatically generated stoplist of size 147 that contains the above listed 10 words of the same study (

see Appendix A.1, A.2). In addition, we extend this stopwords list manually and obtain a list that contains 217 words (see Appendix A.3).

Stemming

We tested three stemming methods in obtaining vectors used for document description. They are (1) no stemming, so called ‘austriach algorithm,’ (2) first n, n-prefix, characters of each word, and (3) a lemmatizer-based stemmer. There is another stemming algorithm, which is based on the ‘successor variety’ (SV) concept [31]. In the SV approach, the root of a word is determined according to the number of distinct succeeding letters for each prefix of the word to be stemmed can have in a large corpus. The idea is intuitively appealing due to agglutinative nature of the Turkish language. The recent IR work done in [21] shows that the SV-based method provides a performance similar to the n-prefix and the lemmatizer-based methods; therefore, it is not considered in this study.

No-Stemming (NS). The no-stemming (NS) option uses words as they are as an indexing term. The performance with this approach provides a baseline for comparison.

Fixed Prefix Stemming (F5, F6). The fixed prefix approach is a pseudo stemming technique. In this method, we simply truncate the words and use the first n (Fn) characters of each word as its stem; words with less than or equal to n characters are used with no truncation. In this study, we experiment with F5 and F6, which are experimentally shown to give the best performance in IR [21]. The success of this method can be explained with the fact that Turkish word roots are not much affected with suffixes [28].

Lemmatizer-based Stemming (LM). A lemmatizer is a morphological analyzer that examines inflected word forms and returns their dictionary forms. Lemmatizers are much more sophisticated than stemmers. They also provide the type (part of speech, POS, information) of these dictionary forms, and the number and type of suffixes (morphemes) that follow the matched forms [47]. Lemmatizers are not stemmers, since the latter obtains the root in which a word is based; in contrast, a lemmatizer tries to find the dictionary entry of a word.

Being an agglutinative language, Turkish has different features from English. For English, stemming may possibly yield ‘stems’ which are not real words. Lemmatization on the other hand tries to identify the ‘actual stem’ or ‘lemma’ of the word, which is the base form of the word that would be found in the dictionary. Due to the nature of English, sometimes words are mapped to lemmas, which apparently do not have any surface connection as in the case of *better* and *best* being mapped to *good*. However, Turkish does not have such irregularities and it is always possible to find the ‘stem’ or ‘lemma’ of any given word through application of grammar rules in removing the suffixes. In the thesis, we prefer the word ‘stemming’ over lemmatization; as it is more commonly used, and the algorithm we use internally identify the suffixes and remove them in the stemming process.

In the lemmatization process, in most of the cases we obtain more than one result for a word. In such cases, the selection of the correct word stem (lemma) is done by using the following steps [11]:

- Select the candidate whose length is closest to the average stem length for distinct words for Turkish.
- If there is more than one candidate, then select the stem whose word type (POS) is the most frequent among the candidates.

3.1.2 Document Representation

We use vector space model, the most popular one, for document representation in our detection and tracking algorithms. In this approach, each unique word in the document is considered as the smallest unit to express information, followed by *TF.IDF* weighting scheme, i.e. term frequency (*tf*) times inverse document frequency (*idf*) [53]. *TF.IDF* values for each term are computed by the following equation.

$$w(t, \vec{d}) = \left(1 + \log_2(tf, \vec{t}) * \log_2(N_t/n_t)\right)$$

In this formula, $w(t, \vec{d})$ is the weight of term t in document (vector) $vecd$; $\log_2(tf, \vec{t})$ is the number of occurrences of term t in document d ; $\log_2(N_t/n_t)$ is the IDF (inverse document frequency), where n_t is the number of stories in the collection that contains one or more occurrence of term t up to the newcomer, and N_t is the number of accumulated stories so far in the collection. Hence, n_t and N_t are incrementally computed, similar approach is used by other researchers [66]. We use an auxiliary corpus containing the 2001-2004 news stories, about 325,000 documents of Milliyet Gazetesi [21] to create IDF statistics of the retrospective corpus, and update the IDF values with each incoming story.

3.1.3 Document Comparison

A similarity value is computed while comparing an incoming story with a topic or a story. Several similarity measures have been used in TDT [8, 15]. In this thesis, we provide a detailed experimental evaluation with two of them: cosine similarity measure [53] and another one which is derived from the cover coefficient (CC) concept [10, 17, 18, 22]. The first one is the most frequently used similarity measure in TDT. The second one is CC concept, which is originally introduced for clustering and based on a probabilistic model. We use these two measures separately and also fuse their result in a simple way. We hypothesize that by fusing their results we will be able to improve the system performance since they are based on different concepts: one is a vector-space representation of stories; the other is originated as a probabilistic measure. We analyze their results in detail.

Cosine Similarity Measure

In the vector space model, the cosine similarity measure is the cosine of the angle between two vectors [52, 53, 60]. The equation is defined as

$$sim(t, d) = \frac{\sum_{j=1}^M w_{jt} * w_{jd}}{\sqrt{(\sum_{j=1}^M w_{jt}^2) * (\sum_{j=1}^M w_{jd}^2)}}$$

where $sim(t, d)$ is the cosine similarity value between document t and d is, w_{jt} is the weight of word j in document t , w_{jd} is the weight of word j in document d ,

$\sqrt{(\sum_{j=1}^M w_{jt}^2) * (\sum_{j=1}^M w_{jd}^2)}$ is the normalization factor. When document t and d are equal, the cosine similarity value will be one.

Cover Coefficient-based Similarity Measure

According to the cover coefficient (CC) concept, the similarity value between two documents d_i and d_j in a collection of m documents defined by n terms is calculated as follows [17, 22].

$$c_{ij} = \alpha_i \cdot \sum_{k=1}^n d_{ik} \cdot \beta_k \cdot d_{jk} \quad (1 \leq i, j \leq m)$$

Where

$$\alpha_i = \left[\sum_{l=1}^n d_{il} \right]^{-1}, \quad \beta_k = \left[\sum_{l=1}^m d_{lk} \right]^{-1}$$

In this formula c_{ij} ($1 \leq i, j \leq m$) indicates the probability of selecting any term of d_i from d_j . In CC, d_{ik} indicates the number of occurrences of term t_k in document d_i (a similar definition applies to d_{jk}). In the experiments, we tried various options with the CC concept for finding the similarity between the newest document and by only considering the members of the sliding time-window, which is defined in the next section. These are by a) only considering the window documents in the calculation of the β values, b) using the 2001-2004 news stories of [1] in an incremental fashion (i.e., in this case it is assumed that the arriving documents are added into this retrospective collection) in the calculation of the β values, and c) a variant of option-b that involves incremental calculation of the β values but by using the following modified α_i and β_k for smoothing their effects on c_{ij} values.

$$\alpha_i = \left[\log \sum_{l=1}^n d_{il} \right]^{-1}, \quad \beta_k = \left[\log \sum_{l=1}^{\text{entirecollection}} d_{lk} \right]^{-1}$$

We take their logarithms of the α_i and β_k values in order to not to overly diminish the d_{ik} and d_{jk} values if the α_i and β_k values are too large. By this way, for example, when the value of β changes from 10^1 to 10^3 , its effect, according to the original formula, is 100 times higher; however, with the logarithmic formula the effect is only 3 times more. Similar normalization approaches are used in other

similarity measures such as INQUERY [33]. In the experiments, the best results are obtained with the option-c and we present the results associated with this option.

Combining Cosine and CC Similarity Scores for NED

We also explore the effects of combining approaches, where the decision of both similarity scores are simply combined using ‘xor’ or ‘and’ function. The reason behind the exploring that we have a hypothesis that the true event, is missed by cosine similarity measure, might be detected by the decision of CC similarity measure. In addition, the false alarm, which is produced by cosine similarity measure, can be decreased by the CC measure. The motivation behind combining two different similarity measure scores are also experimentally discussed in Section 4.3.4. As explained in the next section we use similar combination methods in TT.

The combination of the cosine and CC similarity scores for an incoming story d follows.

$$\text{Let } x = \max_{d_k \in \text{window}} (sim_{\text{cosine}}(d, d_k)) \text{ and } y = \max_{d_k \in \text{window}} (sim_{\text{cc}}(d, d_k))$$

1. and-combination: if $x < \theta_{\text{cosine}}$ and $y < \theta_{\text{cc}}$, then d is labeled as the first story of a new event.
2. or-combination: if $x < \theta_{\text{cosine}}$ or $y < \theta_{\text{cc}}$, then d is labeled as the first story of a new event.

The thresholds θ_{cosine} and θ_{cc} are obtained by training. Other researchers use similar approaches. For example, [32] use a statistical model for combining the results of different similarity measures.

3.1.4 Time Window

In TDT, a newly arrived story, which is not related to any of the previous

stories, is assumed to be the first story of a new event. The relatedness is determined by using a similarity measure. For this purpose, the newest story is compared with all of the previous stories to decide if it is different (dissimilar) from them. However, this process is computationally expensive. Therefore, we use a sliding time-window (see Figure 3.1), in which we only keep the stories of a certain number of most recent days (i.e., 24 hours) and compare the newcomer only with them [40, 48, 66]. In the experiments each 24 hours is approximated by average number of stories per day (aspd) and for each 24 hours we keep the most current aspd number of stories in the time-window. When a new story arrives the oldest story leaves the window.

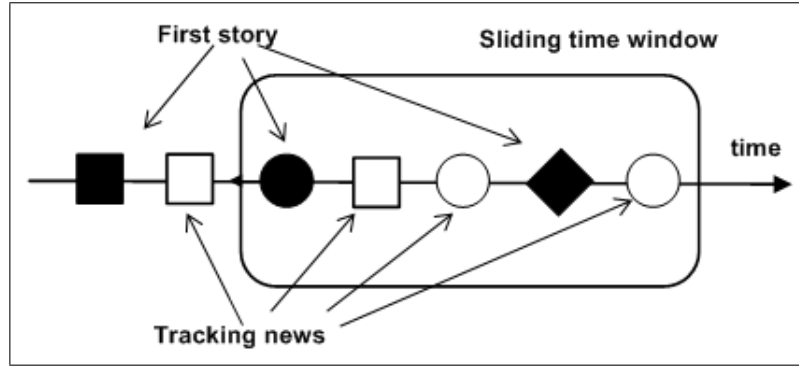


Figure 3.1: Sliding time-window approach in TDT.
(Different shapes represent different events.)

This approach is based on the assumption that the stories related to an event are close to each other in terms of occurrence time.

3.1.5 Novelty Decision

In the detection, novelty threshold is used (*Yiming Learning approaches for detecting and tracking news events 1999*). According to this approach, the flag of ‘NEW’ is assigned to the newest story d if its confidence score, the maximum similarity between d and the individual members of the time-window, is below a pre-determined threshold (t). Otherwise, it is labeled ‘OLD’. The confidence

score for this decision is defined as follows.

$$score(d) = \max_{d_k \in window} (sim(d, d_k))$$

3.1.6 New Event Detection Algorithm

Our new event detection system, where each news story processed sequentially, is depicted in Figure 3.2:

1. Pre-process each story and prepare document vector. If necessary do word stopping and then word stemming.
2. Use the terms of d to update IDF statistics.
3. Compute *TF.IDF* scores of all terms
4. Select the highest weighted n terms for its vector representation.
5. Calculate the similarity scores between the current story and existing news stories in the time-window.
6. If the maximum confidence is below pre-determined threshold then label the story as *NEW*, otherwise label it as *OLD*.
7. Add the story into the time-window and slide time-window, that is, remove the oldest story from the window.

Figure 3.2: The NED algorithm used in the thesis.

3.2 Topic Tracking

The TT task uses N_t (usually 1 to 4) number of sample stories about a given topic t and aims to find stories on that topic ('on-topic' stories) in a news stream. In TT, there would be several topics tracked at the same time and each one is 'tracked separately and individually.' This means that the decision made for one topic does not affect decisions made for the other topics [44]. TT is similar to information filtering, but different from filtering in TT users provide no feedback about the correctness of the system decisions [7].

3.2.1 Adaptive Tracking Approach

In order to catch the appearance of new lexical features for a topic as it evolves in time, we utilize an adaptive approach in our topic tracking. In adaptive approach, the tracking is constructed using static approach and then topic centroid is updated on-line with features from incoming relevant stories. In this approach, we use two different estimated thresholds. For adaptive tracking during standalone use of similarity measures, if the similarity between the arriving story and the topic is above the pre-determined first threshold (for CC, for cosine) is labeled on-topic and after that if the similarity is greater than the pre-defined second threshold , the highest weighted n terms in the story are used to update the topic description. Therefore, adaptation means changing the topic description according to newly found tracking stories. After that, the TF.IDF weights of all terms in the topic are recomputed, and then we re-constructed the topic centroid by selecting n terms with the highest weight. In the combination methods, a similar approach is used.

3.2.2 Combining Cosine and CC Similarity Scores for Tracking

For TT in addition to the standalone use of the CC and cosine similarity measures, we also use the and- and or-combination methods similar to the ones defined for NED (in the following x and y , respectively, indicate the CC and cosine similarity of the incoming story to the topic).

1. and-combination: if $x > \theta_{\cosine}$ and $y > \theta_{cc}$, then the incoming story d is labeled as a tracking story of t .
2. or-combination: if $x > \theta_{\cosine}$ or $y > \theta_{cc}$, then the incoming story d is labeled as a tracking story of t .

3.2.3 Topic Tracking Algorithm

In TT the definition of a topic t is done by using a sample document vector, and each incoming story (d) is considered one by one. The static Topic Tracking algorithm is depicted in Figure 3.3.

1. Pre-process each story and prepare document vector. If necessary do word stopping and then word stemming.
2. Use the terms of d to update IDF statistics.
3. Compute $TF.IDF$ scores of all terms
4. Select the highest weighted n terms for its vector representation.
5. Compute the confidence score ($sim(t, d)$) between current story (d) and targeted topic (t)
6. if the confidence score is above a pre-defined threshold θ_{on} (θ_{cc} for CC and θ_{cosine} for cosine), then it is decided that d is on-topic, otherwise d is classified as off-topic

Figure 3.3: The static TT algorithm used in the thesis.

In the adaptive approach, there is also adaptive threshold (θ_{adapt}) in order for deciding which story modify the targeted topic cluster. The topic tracking with adaptive approach is shown in Figure 3.4.

1. Pre-process each story and prepare document vector. If necessary do word stopping and then word stemming.
2. Use the terms of d to update IDF statistics.
3. Compute $TF.IDF$ scores of all terms
4. Select the highest weighted n terms for its vector representation.
5. Compute the confidence score ($sim(t, d)$) between current story (d) and targeted topic (t)
6. If the confidence score is above a pre-defined threshold θ_{on} (θ_{cc} for CC and θ_{cosine} for cosine), then it is decided that d is on-topic, otherwise d is classified as off-topic
7. If the confidence score is above θ_{adapt} , then we update the vector representation of t using its current terms along with those of the current document d .

Figure 3.4: The adaptive TT algorithm used in the thesis.

Chapter 4

Experimental Evaluation

4.1 Data Set

The BilCol2005 test collection developed by the Bilkent Information Retrieval Group. The collection contains 209,305 documents from the entire year of 2005 and involves several events in which eighty of them are annotated by humans [20, 46]. The annotated topics with the number of tracking stories and their life span are depicted in the Appendix Table B.1 and Table B.2. It is observed that the average topic life is 92 (median: 59, minimum: 1, maximum: 357) days.

For experimental evaluation, we divide the test collection into two parts: training and test sets. For this purpose, the first eight months served as the training data, and the last four months as the test data. Such a division gave us the opportunity of keeping most of the tracking stories together with their corresponding first stories. For example, dividing the data set into two six-month periods (January to June, and July to December) does not give us that opportunity (see Figure 4.1). X axis in the figure goes from Jan.1 to Dec. 31, 2005. Each horizontal position represents a different event and there are 80 events. The gray level is proportional to the number stories on that day, darker gray spots indicate more stories. Days with 10 or more stories are shown with the same gray color.

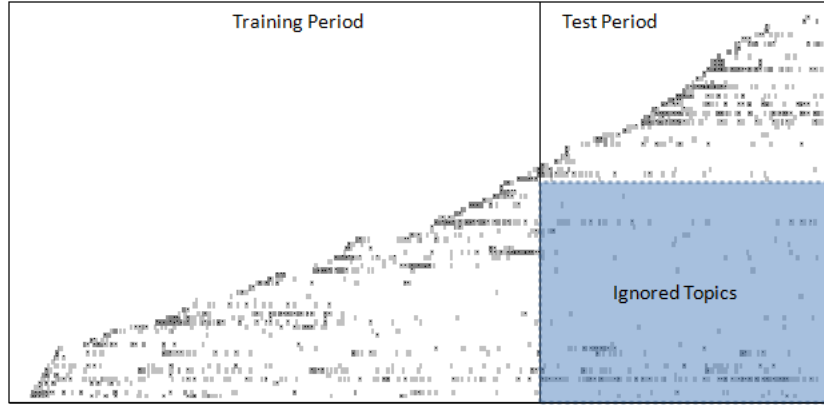


Figure 4.1: The distribution of BilCol2005 topic stories among the days of 2005.

All together, we have 80 topics. For two topics used for training, there is considerable number of news stories in the period that corresponds to the test data. For these two topics, their first stories in the test set section are used as the first stories of the two new events. By this way all-together in the train and test sets there are 82 topics (i.e., two more than original 80 topics). As seen in Table 4.1, the average number of news per topic in the train and test sets are approximately the same 67.16 (3,358/50) and 71.50 (2,288/32) stories.

4.2 Evaluation Methodology

The most common evaluation measures in TDT are false alarm (FA) and miss rate (MR). The definitions of these effectiveness measures are defined as follows.

1. False Alarm (FA) = number of tracking news labeled as new event / total number of tracking news,
2. Miss Rate (MR) = number of new events labeled as tracking news / number of all new events.

These are both error measures and the goal is to minimize them. In the ideal case they are both equal to 0.

FA and MR are shown with a curve by using false alarms and miss rates gathered from various similarity threshold values that are used for decision-making [7, 30]. This curve is defined as detection error trade-off (DET) curve, which is similar to traditional ROC (receiver operating characteristic or relative operating characteristic) curve [42]. DET curves provide a visualization of the trade off between FA and MR. They are obtained by moving thresholds on the detection decision confidence scores. In obtaining the overall system performance, we use the topic-weighted approach that assigns the same importance to all topics independent of their number of tracking stories. This approach is commonly used in the literature and more preferable than story-weighted estimates [30].

DET curves provide detailed information; however, they may be difficult to use for comparison. For this reason, in TDT, there is another effectiveness measure, detection cost function, C_{Det} , which combines FA and MR and yields a single value for measuring the effectiveness [13, 30]. The detection cost function is defined as follows.

$$C_{Det} = C_{Miss} * P_{Miss} * P_{Target} + C_{FA} * P_{FA} * (1 - P_{Target})$$

where

1. $C_{Miss} = 1$ and $C_{FA} = 0.1$ are the costs of a missed detection and a false alarm, and they are pre-specified,
2. $P_{Target} = 0.02$, the *a priori* probability of finding a target as specified by the application,
3. P_{Miss} : miss probability (rate) determined by the evaluation result,
4. P_{FA} : false alarm probability (rate) determined by the evaluation result.

These pre-specified numerical values are consistently used in TDT performance evaluation [29, 30].

The formula given for C_{Det} has a dynamic range of values and is difficult to use for relative comparison. For this reason, a normalized version of C_{Det} is used for comparison. It has a more meaningful range.

$$(C_{Det})_{norm} = \frac{C_{Det}}{\text{Minimum}\{C_{Miss} * P_{Target}, C_{FA} * (1 - P_{Target})\}}$$

The values obtained by this formula generally 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’ [29, 30]. In our work we use the normalized version of the C_{Det} formula.

To evaluate performance, the stories are sorted according to their scores, and a threshold sweep is performed. All stories with scores above the threshold are declared 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 number. The threshold that results in the least cost is selected as the optimum one [35]. Different NED systems are compared based on their minimum cost. This minimum cost is depicted as $\min C_{Det}$. $\min(norm(C_{det})) = \min\{norm(C_{det})\}$ $norm(C_{det}) \in S$ where S is the set of all minimum normalized cost values calculated by in each threshold value using threshold sweep approach.

Moreover, we also used statistical one-tailed paired t-tests over topics’ C_{Det} value using alpha level of 0.05 for significance level. In the statistical test, the difference between pair samples are assumed to be a random sample from a normal distribution with mean zero and unknown variance against the alternative that the mean is not zero. Before doing the statistical tests, the C_{Det} values, which were identified as potential outliers, are eliminated. An potential outlier is defined as an observation that is more than 2.5 standard deviations above or below the mean. In other words, we first compute the differences for each pair and we extract the pair whose value is more or less than 2.5 standard deviations from the mean.

4.3 New Event Detection Results

4.3.1 Vector Length, Stemmer, and Window Size Tuning

Table 4.1: Distributions of stories among training and test sets

Corpus Type	Time Span	No. of Topics	No. of Docs	No. of Relevant Docs
Training	01.01.2005 - 08.31.2005	50	141,910	3,358
Test	09.01.2005 - 12.31.2005	32	67,395	2,288

Numerous combinations are possible for the document vector length, stemmer, and window size parameters. For TDT it would be reasonable to choose a window size that would give us an opportunity of finding the tracking stories of topics. For the majority of the event stories, the average time difference among the stories of a topic is less than or equal to 12 days. For selecting a suitable ‘document vector length and stemmer’ combination, we used the 12-day sliding time-window size and analyzed the system performance. Table 4.2 shows the performance in terms of minimum (Min.) C_{Det} values with the CC and cosine similarity measures with different (document) vector length and stemmer combinations. In general, stemming approaches provide a better performance than no stemming (NS). For example, with the cosine measure F5, F6, and LM always outperform NS in all vector length cases. Again with the same measure, using all terms of documents and the lemmatizer-based stemmer (LM) provides the best performance. Furthermore, in 15 of the 16 different vector lengths LM provides the best performance. With the cosine measure, observing the best performance with all terms is consistent with the results of other researchers [8].

In terms of CC measure, F6 with the vector length of 60 provides the best performance. Furthermore, the simple truncation method of fixed prefix is better than LM in 13 of the 16 cases. In fact, F6 yields the best performance, i.e., the least minimum normalized cost in detection, for most vector lengths (10 cases out of 16 different vector lengths F6 provides a lower cost than that of NS, in 14 cases it is better than F5, and in 13 cases it is better than LM).

The best case for each stemmer is in bold and the best case of each

Table 4.2: Min. C_{Det} values for CC and cosine measures for ned with various vector lengths and stemmers

Vector	Cover Coefficient (CC) Performance				Cosine Performance			
	NS	F5	F6	LM	NS	F5	F6	LM
10	0.8996	0.8957	0.8863	0.8743	0.9517	0.7683	0.7555	0.7590
20	0.8092	0.7292	0.7462	0.6842	0.7669	0.7000	0.7437	0.6204
30	0.7268	0.6657	0.6408	0.6751	0.7476	0.6352	0.7115	0.6220
40	0.6849	0.6714	0.6198	0.6433	0.7222	0.6284	0.6562	0.5977
50	0.6841	0.6361	0.6097	0.5950	0.7696	0.6455	0.6485	0.5860
60	0.6411	<u>0.5929</u>	<u>0.5730*</u>	0.5982	0.7695	0.6312	0.6771	0.5974
70	0.6325	0.6358	0.5926	0.6235	0.7635	0.6277	0.6692	0.6019
80	0.6300	0.6490	0.5973	0.6740	0.7274	0.6354	0.6581	0.6115
90	0.6255	0.6462	0.6151	0.7408	0.7248	0.6349	0.6405	0.5916
100	0.6043	0.6512	0.6586	0.7610	0.7025	0.6347	0.6337	0.5849
120	0.6325	0.7306	0.6792	0.8001	0.7007	0.6326	0.6661	0.5972
140	0.6686	0.7637	0.7482	0.8142	0.7210	0.6357	0.6572	0.5913
160	0.7093	0.7639	0.7599	0.8542	0.7411	0.6313	0.6516	0.5965
180	0.7570	0.7872	0.7610	0.8658	0.7225	0.6313	0.6504	0.5912
200	0.7917	0.7857	0.7728	0.8746	0.7155	0.6331	0.6458	0.5909
All	0.8012	0.8943	0.8512	0.9444	0.6872	0.6101	0.6176	<u>0.5777*</u>

similarity measure is underlined.

While using CC, $F6$ provide the best C_{Det} value with the vector length of 60. Therefore, we hypothesize that the use of $F6$ provide better results than other stemming approach while using the vector length of 60. The one-tailed statistical test results also indicate that while using CC similarity measure in detection, $F6$ is significantly better than NS ($p < 0.01$). The experimental results with CC show that in the Turkish TDT, depending on the used similarity measure, a simple word truncation stemming method can compete with a sophisticated stemming approach that pays attention to the morphological structure of the language. Moreover, while using cosine similarity, LM with all terms provide best C_{Det} cost. As using this vector length in both cases, the statistical results show that LM is found to be strongly better than NS stemmer ($p < 0.001$).

We also evaluated the effect of stemming. In this experiment, the vector length of 60 words is used for CC whereas all terms are used in the vector for cosine. Test results are depicted in the following table:

The one-tailed statistical test results show that when cosine similarity is used, LM is significantly better than other stemmers ($p < 0.05$). However, when CC

Table 4.3: Effects of stemmer on effectiveness with CC and cosine measures

Stemmer	Cover Coefficient			Cosine Similarity		
	Training Results		Test	Training Results		Test
	Min C_{Det}	Threshold	C_{Det}	Min C_{Det}	Threshold	C_{Det}
NS	0.6411	0.199	0.7731	0.6872	0.099	0.7280
F5	0.5929	0.252	0.7637	0.6101	0.166	0.6060
F6	0.5730	0.254	0.6476	0.6176	0.155	0.6627
LM	0.5982	0.290	0.7672	0.5777	0.200	0.4947

similarity is used, $F6$ is significantly better than $F5$ and NS ($p < 0.05$) but slightly better than LM ($p < 0.10$).

In the rest of the paper, unless otherwise specified, we use the stemmer and vector length combinations of ‘F6-60 terms’ and ‘LM-all terms,’ respectively, for the CC and cosine similarity measures.

4.3.2 Effects of the Stopword list

The results presented in Table 4.4 show that stopwords lists and their sizes have an influence on the effectiveness and as we increase the stopwords list size, positive influence of word stopping with respect to the baseline, no stopping, increases. (Note that in these experiments time-window size, stemmer, and document vector length are kept the same and only the stopwords list is varied. The training Min C_{Det} values in the Table 4.4 can also be seen in Table 4.2. The experimental conditions are as specified in that table that corresponds to these C_{Det} values).

Table 4.4: Effects of stopwords list size on effectiveness with CC and cosine measures

Stop List	Cover Coefficient			Cosine Similarity		
	Training Results		Test Results	Training Results		Test Results
	Min C_{Det}	Threshold	C_{Det}	Min C_{Det}	Threshold	C_{Det}
0	0.7149	0.256	0.7622	0.7050	0.226	0.5809
10	0.6512	0.246	0.7199	0.7052	0.211	0.5870
147	0.5895	0.244	0.7286	0.5958	0.200	0.5292
217	0.5730	0.254	0.6476	0.5777	0.200	0.4947

Moreover, we also conducted one-tailed paired t-test on the training and the test topic’s C_{Det} values. The statistical results on the training and the test corpus indicate that using the longest stopwords list statistically yields lower C_{Det} values

than using no stopwords for each similarity measure ($p < 0.05$ for all cases). As a result, the use of such stopwords increases NED effectiveness. On the other hand there is no statistically significant difference between using the longest stop list and using the list with 147 words, but we use the longest stopwords list in the remaining experiments because it provides the best performance.

In the recent IR research, it is observed that word stopping does have no noticeable influence on the retrieval performance in Turkish [21]. However, as presented above it is not the case with TDT. This can be explained by the fact that normally (or more specifically in information retrieval) stopwords are not used in the queries. However, in TDT applications, news articles are used like a query. There are many such terms in news articles. For example, in the training data set, on the average in each document there are 135 unique words and 14 of them appear in the largest stopwords list.

4.3.3 Effect of Window Size

After these observations, we reconsidered the effect of the window size and obtained the performance with several different window sizes. The results are shown in Figure 4.2.

We observe that window size of 12 and 14 days are the best choices for cosine and CC similarity measures since they provide the minimum C_{Det} for these similarity measures, respectively. For cosine 12 and 18 days provide the same performance; for efficiency, we prefer a smaller window size. In the remaining experiments, we use these window sizes for the CC and cosine measures. The fact that our window sizes come out so similar suggest that we are simply learning from noise; however, we still use these different window sizes for these two different similarity measures.

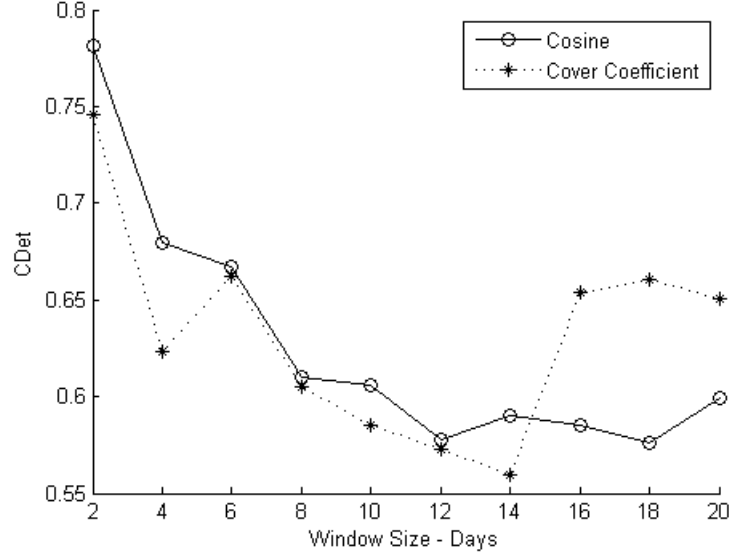


Figure 4.2: Min. C_{Det} with different time-window sizes for CC and cosine measures.

4.3.3.1 Confidence Scores

In this study, the main purpose of our effort is correctly identifying new events based on confidence scores, i.e., the maximum similarity between the most recent incoming story and the current members of the time-window. We examined the distribution of the confidence scores of documents that correspond to new events (first stories) and tracking stories for the CC and cosine similarity measures. These scores are plotted in Figure 4.3. Remember that a new story, which is different from, or not similar to, current time-window members correspond to a new event.

Figure 4.3 shows that a large portion of first (new) stories have a small confidence score values, i.e., values less than about 0.2, whereas a small fraction of (old) tracking stories have low scores, e.g., scores less than 0.2. The CC confidence scores of a significant number of tracking stories (compared to first stories) are close to 1. (Note that the CC scores can have a value greater than 1, in Figure 8 for a simpler presentation such values are converged to 1.) In addition to this,

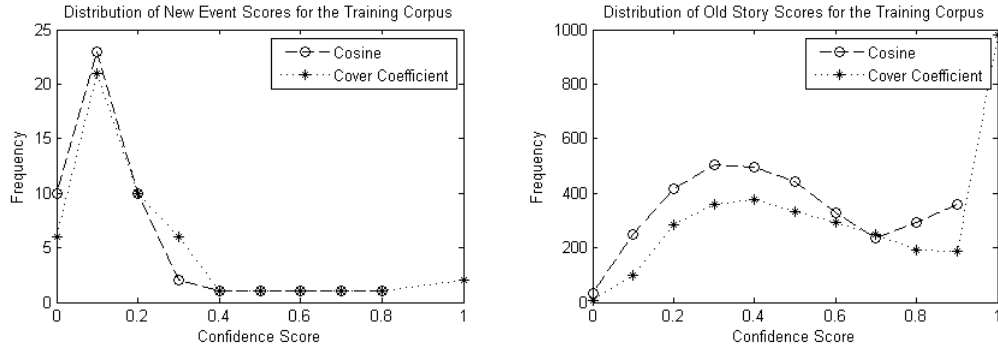


Figure 4.3: Confidence scores of first (left figure) and tracking (right figure) stories. In CC, the confidences score 1 corresponds to a CC value 1 or higher.

the cosine confidence scores of a relatively large number of first stories (compared to tracking stories) are close to 0. These observations show that the distribution pattern of the confidence scores of these two measures are similar to each other.

4.3.4 Intuition Behind our Combination Methods: Decision Consistency of CC and Cosine Similarity measures

We first analyzed the decision consistency of these two measures. The left Venn diagram of Figure 4.4 shows consistency in the correct NED decisions (true matches or hits) when the threshold values obtained during training are used on the training data set for decision-making. It shows that these two measures agree on 23 decisions jointly and 7 cases are only detected by cosine and 6 cases are only detected by CC and all together we have 36 correct decisions out of 50. The consistency between CC and cosine decisions in correct decisions is about 78% (23/30 and 23/29). The right Venn diagram shows the case for the false alarms generated by tracking news during NED. It shows that the cosine and CC measures, respectively, incorrectly detect 110 and 127 tracking stories as new events and among these they have 50 common false drops. Their agreement in making

wrong decisions is about 42%.

The observations indicate that these measures are more consistent in correct decisions and less consistent in wrong decisions. This intuitively implies that potentially they may correct each other if both of them are used together for decision-making. We can interpret these observations as follows. 1) The left illustration suggests that if we use these two measures together, when cosine ‘or’ CC detects a new event, and if we accept that decision we would be able to increase the number of true new event matches. 2) The right illustration suggests that if cosine ‘and’ CC decisions agree and if we make a decision when both of them agree then we would be able to decrease the number of false alarms.

These observations give us the motivation for combining the results of two different similarity functions for decision making in NED and TT. Therefore, we describe a combined approach that employs the confidence scores coming from these two measures for the detection of new events by ‘or’ing their decisions. We also try an ‘and’ing version of this combined approach. As explained above we intuitively expect that, the joint use of these scores coming from two different similarity measures would increase the number of correct decisions and improve the NED accuracy. This is conceptually similar to rank aggregation used in data fusion [45].

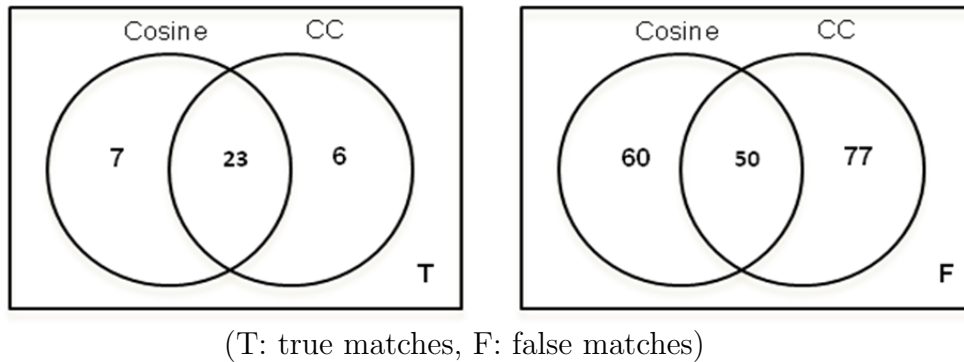


Figure 4.4: Consistency of CC and cosine decisions in NED during training for 50 events.

4.3.5 Training Results

The training performance is shown in Figure 4.5 by using a DET curve (NIST, 2008). At smaller false alarm values CC measure provides a slightly better performance than cosine, and for false alarm values greater than (5%) the reverse is true. In terms of Min. C_{Det} , CC provides a slightly better performance than cosine. During testing, for each similarity measure we use the similarity threshold value that gives best performance, i.e., Min. C_{Det} value, during training.

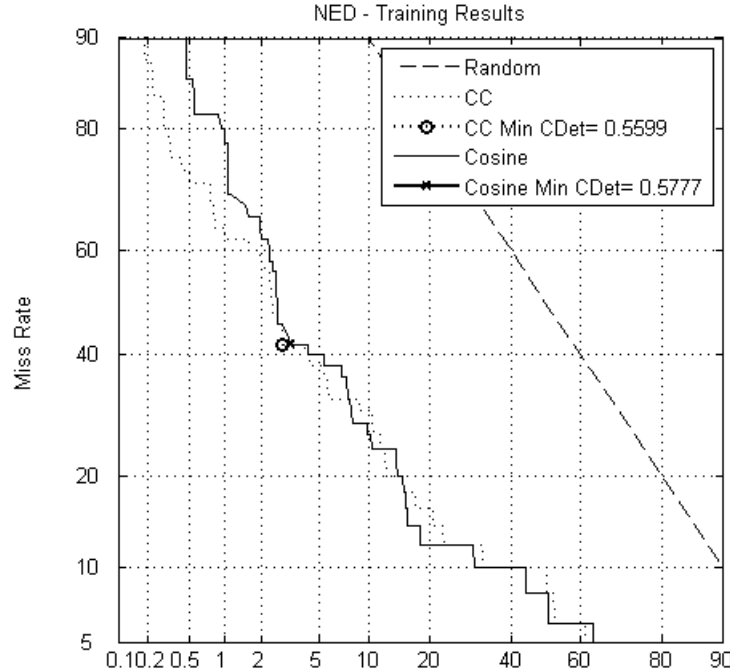


Figure 4.5: NED training performance of CC and cosine measures.

It is observed that in training the ‘or-combination’ method gives the best performance in NED because we miss less new events than the other approaches. The optimum similarity thresholds values are 0.254 and 0.200 for CC and cosine measures, respectively. In addition, we observe that optimum thresholds for the and- and ‘or-combination’ methods are the same as the case with the standalone use of these similarity measures. When the similarity measures are used alone then in that case CC provides a better performance than cosine. The optimum training results for each new event detection approach are given in Table 4.5. The best performance with the standalone and combined versions are in bold.

Table 4.5: NED training results

Similarity Measure	Miss Rate	False Alarm	Min. C_{Det}	Threshold (θ_{on})
CC	0.4200	0.0285	0.5599	$\theta_{cc} = 0.254$
cosine	0.4200	0.0322	0.5777	$\theta_{cos} = 0.200$
and-combination	0.5600	0.0136	0.6268	$\theta_{cc} = 0.254, \theta_{cos} = 0.200$
or-combination	0.2800	0.0471	0.5108	$\theta_{cc} = 0.254, \theta_{cos} = 0.200$

(Best standalone and combination result is in bold, best case is underlined.)

4.3.6 Evaluation Results

Using the training thresholds we evaluate each approach using the 32 test topics. The test results are provided in Table 4.6. During testing, C_{Det} values are obtained by using the threshold values that are determined during training.

Table 4.6: NED test results

Similarity Measure	Miss Rate	False Alarm	C_{Det}
CC	0.5313	0.0214	0.6359
cosine	0.3750	0.0244	0.4947
and-combination	0.6250	0.0103	0.6757
or-combination	0.2813	0.0355	0.4550

(Best standalone and combination result is in bold, best case is underlined.)

From these observations, the use of cosine provides a better performance than the use of CC. However, the use of CC has a positive impact when we use it in the use of ‘or-combination’. During testing the ‘or-combination’ provides the best performance and percentage reduced cost of -8.03% (0.4550 vs. 0.4947) and -28.45% (0.4550 vs. 0.6359) with respect to the standalone use of cosine and CC in testing. We also conduct one-tailed paired t-test over test topic’s C_{det} . The results show that ‘or-combination’ is significantly better than ‘and-combination’ and CC ($p < 0.05$). However, there is no statistical evidence on the difference between ‘or-combination’ and cosine ($p = 0.23$).

4.4 Topic Tracking Results

4.4.1 Determining Vector Lengths and Stemmer

During adaptive tracking the definition of t can be seen as a topic centroid, since it is updated according to the newly tracked documents as defined below. We found that the vector length of 60 ($n = 60$) provides the best performance for the CC measure for the description of documents and topics. This is true both for static and adaptive tracking. For the cosine measure, the use of all document terms gives the best performance during static tracking. However, during adaptive tracking this approach could not be used since if there is no limit then the number of topic centroid terms would increase in an uncontrollable manner. Therefore, during adaptive tracking for the cosine measure, for the document vectors and topic centroid definition we only use 100 ($n = 100$) terms with the highest *TF.IDF* weight. Experimentally it has been determined that the use of 100 terms provides the best performance. Note that during NED training for CC the vector length of 60 gives the best performance and 100 gives the second best performance for the cosine measure (see Table 4.2). A similar approach for adaptive tracking is also used in [25].

4.4.2 Training Results

We performed several experiments to define the best threshold(s) for a given case. Note that during the combination methods we have two thresholds to determine; therefore, many possible value combinations. The θ_{on} thresholds for static (non-adaptive) topic tracking are shown in Table 4.7.

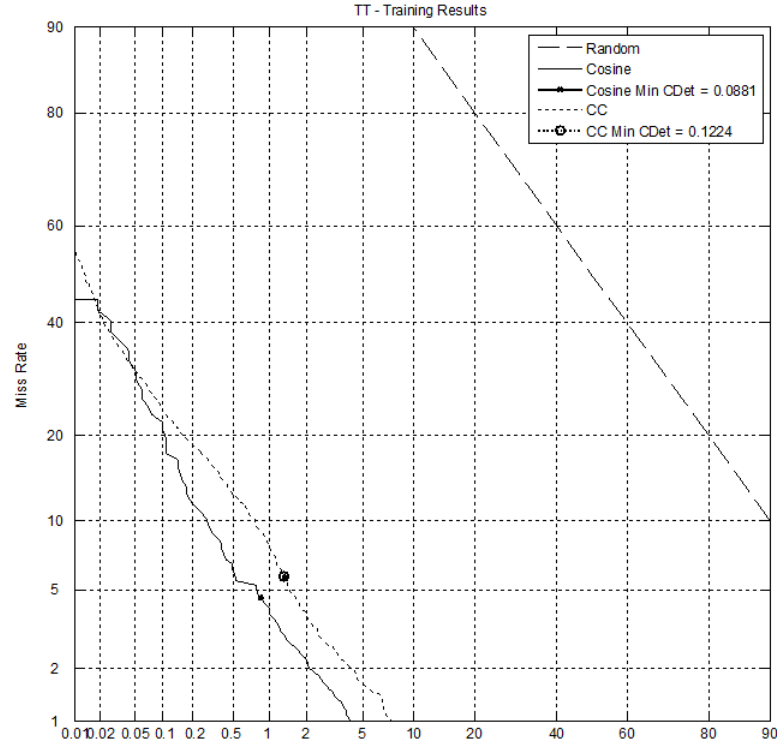
In the static training results, we observed that the combined system with the ‘and-combination’ provides the least C_{Det} values in static TT. However, probably in practice, we might only sense some performance difference between CC and the other methods, since the performance with cosine, ‘and-’ and ‘or-combinations’ are close to each other. The use of CC, cosine and ‘or-combination’ yields more

Table 4.7: Static TT training results

Similarity Measure	Miss Rate	False Alarm	Min. C_{Det}	Threshold (θ_{on})
CC	0.0580	0.0130	0.1224	$\theta_{cc} = 0.097$
cosine	0.0462	0.0085	0.0881	$\theta_{cos} = 0.099$
and-combination	0.0425	0.0083	<u>0.0824</u>	$\theta_{cc}=0.045$ $\theta_{cos}=0.094$
or-combination	0.0462	0.0085	0.0881	$\theta_{cc}=0.335$ $\theta_{cos}=0.099$

(Best standalone and combination result is in bold, best case is underlined.)

33.67%, 6.47%, 6.47% C_{Det} than the use of ‘or-combination’. These are all statistically significant improvements ($p < 0.05$ for all case). Figure 4.6 shows the training performance during static topic tracking. In this figure, the difference between CC and cosine performances is easy to see.

Figure 4.6: Static TT training Min. C_{Det} of CC and cosine similarity ($N_t = 1$).

In comparing the performance of the adaptive TT, let training thresholds denote as $\theta_{on}(\theta_{cc}, \theta_{cosine})$ and denote adaptive thresholds as θ_{adapt} (again for CC and cosine) obtained for adaptive TT are shown in Table 4.8. It is observed that the ‘and-combination’ technique statistically better than CC, cosine ($p < 0.05$, for

both case) but slightly better than the use of ‘or-combination’ ($p < 10$). In Table 4.8, best standalone and combination result is in bold, best case is underlined.

Table 4.8: Adaptive TT training results

Similarity	Miss R.	False A.	Min. C_{Det}	θ_{on}	θ_{adapt}
CC	0.0540	0.008	0.0910	$\theta_{cc}= 0.085$	$\theta_{cc}= 0.400$
cosine	0.0320	0.007	0.0650	$\theta_{cos}= 0.076$	$\theta_{cos}= 0.300$
and-combination	0.0260	0.007	<u>0.0580</u>	$\theta_{cc}= 0.027$ $\theta_{cos}= 0.069$	$\theta_{cc}= 0.400$ $\theta_{cos}= 0.300$
or-combination	0.0330	0.006	0.0650	$\theta_{cc}= 0.149$ $\theta_{cos}= 0.080,$	$\theta_{cc}= 0.400$ $\theta_{cos}= 0.300$

(Best standalone and combination result is in bold, best case is underlined.)

4.4.3 Test Results

We use Turkish evaluation corpus with 32 topics to test the performance of all TT approaches by using the threshold values obtained in the training experiments. The static TT test results are first presented in Table 4.9.

Table 4.9: Static TT test results

Similarity Measure	Miss Rate	False Alarm	C_{Det}
CC	0.0355	0.0188	0.1277
cosine	0.0494	0.0071	0.0842
and-combination	0.0445	0.0071	<u>0.0791</u>
or-combination	0.0494	0.0071	0.0843

(Best standalone and combination result is in bold, best case is underlined.)

The ‘and-combination’ method in static topic tracking provides the least topic weighted C_{Det} cost. The standalone use of CC yields statistically the worst performance, i.e., highest C_{Det} value of 0.1277 ($p < 0.01$). In addition, the statistical test results also indicated that the use of ‘and-combination’ increase significantly the system performance compared to cosine, CC and ‘or-combination’ measure.

The test results for adaptive TT are then presented in Table 4.10. It is observed that the standalone use of CC gives a performance close to the standalone use of cosine, but not as good. The ‘and-combination’ method gives the least C_{Det} values. The percentage cost reduction provided by ‘or-combination’ with respect to standalone use of these similarity measures is noticeable: it is

Table 4.10: Adaptive TT test results

Similarity Measure	Miss Rate	False Alarm	C_{Det}
CC	0.0580	0.003	0.0699
cosine	0.0350	0.004	0.0563
and-combination	0.0200	0.006	0.0513
or-combination	0.0170	0.006	<u>0.0461</u>

(Best standalone and combination result in bold, best case is underlined.)

-8.88% (0.563 vs. 0.513) and -26.60% (0.699 vs. 0.0513). Even though this increase is noticeable, one-tailed statistical results over topic's C_{Det} values show that there is no significant evidence for difference between 'or-combination' and cosine ($p > 0.10$). Moreover, the test results also indicate that the use of 'and-combination' is slightly better than the use of CC ($p < 0.10$). Furthermore, there is no statistical evidence for difference between 'and-combination' and 'or-combination' ($p > 0.10$). Even though this difference is not significant, the 'or-combination' provides a promising approach; therefore, this 'or-combination' may be preferred to be used in the tracking.

On the other hand, the comparison of static (Table 4.9) and adaptive (Table 4.10) TT test results show that adaptive tracking is statistically more advantageous than static tracking ($p < 0.05$ for each case).

4.5 Chapter Summary

In this chapter, we first defined our large scale Turkish collection corpus used in the new detection detection and topic tracking. The collection corpus is divided into two parts such as training and test corpus. Then, we described the evaluation methodology in order to compare the effectiveness of the approaches described in the chapter. After that, we tuned the parameters, which are used in the approaches, with training corpus so as to get minimum C_{DET} cost. Finally, we evaluated our approaches with their optimum parameters using test corpus. The experimental results suggested that eliminating stopwords from document vector and stemming increase statistically effectiveness of the detection system. Moreover, the combining two confidence scores provide a promising increase in

the detection, especially ‘or-combination’. Furthermore, using ‘or-combination’ provides a statistically significant increase for system performance in the static topic tracking. However, we observed that the use of and- and or-combinations did not improve adaptive topic tracking. Lastly, the use of adaptive approach was observed more statistically advantageous than the static topic tracking.

Chapter 5

Further Experiments

In this chapter, we briefly describe our observations associated with some other approaches that we tried with new event detection. Further experiments on NED with the Okapi similarity measure using the term chronological ranking concept [57] can be seen in [13].

5.1 Influence of other Similarity Measures on NED

For understanding the influence of similarity measure on the TDT performance, we also experimented with five additional similarity measures (Dice, Hellinger, Jaccard, Okapi, and overlap). These measures are commonly used in TDT or IR. For example, Luo et al. use Okapi and Brants et al. use Hellinger for new event detection [15, 40]. The definitions of the other similarity measures (Dice, Jaccard, overlap) can be seen in [12, 52]. In the experiments, we use *TF.IDF* term weights for term selection. We also use them in Dice, Jaccard, and overlap for term weighting. After adding these five measures all together we have seven similarity coefficients. We know that NED and TT performances are related; therefore, we only experiment with NED. Furthermore, C_{Det} values calculated in

NED have a larger scale. Hence, in NED the influence of a similarity measure on performance, if any, is easier to recognize. While performing these experiments we use a) word stopping using the longest stopword list given in the appendix, b) the incremental IDF approach for the selection of the indexing terms, and c) determine the vector length and window size as we have done for the CC and cosine measures in earlier sections. In all cases, the window size is between 12 and 14 days respectively for the CC and cosine similarity measures.

The results provided in Table 5.1 show that basic vector space model with the cosine similarity measure provides the best performance with the lowest C_{Det} value of 0.4948 in testing. If the cosine observation is taken as a reference point, the percentage reduced cost provided by all of the other measures are positive (i.e., rather than reducing, they indeed increase the cost with respect to that of cosine). Okapi, with the C_{Det} value of 0.5249 and the percentage cost increase of 2.1%, provides the second best performance after cosine. After that, the Jaccard measure comes with a cost increase of 4.2%. In five out of seven cases (namely cosine, Dice, Hellinger, Jaccard, and Okapi measures), the best performance is observed with the lemmatizer-based stemmer LM. However, in two cases (CC and overlap similarity measures) the simple term truncation approaches (F5, F6) outperform LM. In four cases out of seven, it is better to use all terms of news stories for indexing.

In addition to individual performance analyses, we also assess the performance when these similarity measures are used in a pairwise fashion. This has already been done for the CC and cosine measures in an earlier section, their results are repeated here for easy comparison. By this way, we have 21 different pairs, and we perform experiments by using both and- and ‘or-combination’s of two similarity measures for improving the performance with respect to the best case of the combined similarity measures. Tables 5.2 and 5.3, respectively, show the performances of different combinations for the and- and ‘or-combination’ methods for the training and test cases. In each cell of these tables there are two C_{Det} values, the upper and lower scores, respectively, correspond to training (in this case it is the minimum C_{Det}) and test cases (C_{Det} value obtained by using the threshold determined by training). For understanding the influence we only consider the

test scores.

For the ‘and-combination’, the lowest C_{Det} (0.4916) value is observed with the cosine-overlap combination. However, for this case the percentage improvement (i.e., percentage reduced cost) is only -0.65% (0.4916 versus the best of the cosine and overlap C_{Det} values when they are used alone: 0.4948). The Okapi-overlap functions with the ‘and-combination’ gives percentage reduced cost of about -5% (0.5012 versus the best of the Okapi and overlap C_{Det} values: 0.5249, again when they are used alone) cost reduction. It is the highest cost reduction with the ‘and-combination’ approach. The other cases of ‘and-combination’ provide a small improvement or decrease the performance.

Table 5.1: NED standalone use of similarity measures training and test performance

Similarity function	Stemmer	Vector Size	Training(Min. C_{Det})	Test (C_{Det})
CC	F6	60	0.5599	0.6359 (28.5)
Cosine	LM	All Terms	0.5777	0.4948 (0.0)
Dice	LM	All Terms	0.5669	0.5165 (10.4)
Hellinger	LM	All Terms	0.6207	0.5638 (14.0)
Jaccard	LM	All Terms	0.5664	0.5154 (4.2)
Okapi	LM	50	0.5424	0.5249 (2.1)
Overlap	F5	30	0.6573	0.7700 (55.6)

We conduct one-tailed paired t-test between cosine and other confidence scores over C_{DET} values of the topics. The statistical results indicate that the use of cosine measure yields lower C_{DET} values than CC and overlap measures ($p < 0.05$ for both cases). However, there are no significant differences between scores of cosine and the other similarity measures ($p > 0.20$ for all cases).

In the case of ‘or-combination’, the lowest C_{Det} value (0.4550) is observed with the CC-cosine combination. For this case the percentage reduced cost is -8.04% (0.4550 versus the best of the CC and cosine C_{Det} value when they are used alone: 0.4948). Cosine-okapi combination provides the second best C_{Det} value of 0.4717, and percentage reduced cost of -4.67%. The other cases of ‘or-combination’ provide a small improvement or decrease the performance.

The positive observations indicate that combining the results of different similarity measures improves the performance in some cases. The CC-cosine ‘or-combination’ that gives the C_{Det} value of 0.4550 (with -8.04% percentage cost reduction) is noticeable.

These results also suggest that in terms of effectiveness, NED in the Turkish language is similar to those of other languages. For example, Zhang et al. provide some state-of-the-art observations for TDT2 and TDT3 corpora [67]. Their NED minimum C_{Det} values obtained for the TDT2 and TDT3 data sets are between ‘0.5089 and 0.6673’ and ‘0.5012 and 0.6308,’ respectively (please refer to Tables 3 and 4 in their work). Our minimum C_{Det} values for the standalone use of the seven similarity measures that we employ in our experiments are between ‘0.5424 and 0.6573.’ Allan et al. provide evidence in the same direction for the Hindi language [6]. They report the minimum C_{Det} value of 0.5608 for NED (see Figure 3 in their work). The performance we obtain in our work is similar to the ones observed in these works. However, we have to remember that different language environments are unlike not only in terms of the language, but also in terms of test collections and methods. Hence, statements such as ours about the system performance is an approximation. Nevertheless, similar to our efforts, Allan et al. compare the Hindi language results with those of the TDT in other languages [6].

We also conduct one-tailed paired t-test on the best results of the ‘and-combination’ and ‘or-combination’. As stated before, we consider the topic’s C_{Det} for the statistics. For the ‘and-combination’ case, the combination of cosine and overlap similarity provide the best test results. The statistical results indicated that there is strong statistical difference between ‘and-combination’ and overlap ($p < 0.001$) scores, however; there is no statistical evidence for difference between the ‘and-combination’ score and cosine similarity scores ($p = 0.11$). For the ‘or-combination’, the combination of CC and cosine yields least C_{Det} cost in test experiments. The statistical results for this solution show that there is a significance difference between ‘or-combination’ and CC but there is no evidence between cosine and ‘or-combination’.

5.2 Effect of Time Penalty Function

We tested a time decaying approach that assigned lower importance to earlier stories in confidence score calculations. By this way, old stories have progressively less influence on the current decisions. Our approach is similar to the one defined in [66]. In the experiments, we tested CC and the cosine similarity measures. With the decaying function, as time goes on, stories have gradually less effect on the current decision score. The confidence score for this decision is calculated with a modified formula, which is described as follows.

$$score(\vec{x}) = \underbrace{\max}_{(\vec{d}_i \in window)} \frac{i}{m} sim(\vec{x}, \vec{d}_i)$$

where $\vec{x}$ represents vector of the current story and $\vec{d}_i$ is i -th story in the window, and $i = 1, 2, 3, \dots, m$. The document d_1 is the oldest story and d_m is the newest. In our baseline NED, each story in the window has a equal influence on the decision. However, this decaying-weighted function provides a smoother temporal proximity than the baseline. Experimentally, we observed that this approach degraded the system performance when we use it with the selected window sizes 12 and 14 days respectively for the cosine and CC measures. We speculate that by training, we are able to choose a proper window size and due to this, the time decay is unnecessary and if used it has a degenerative effect. However, if average time span of typical events cannot be predicted correctly, then in that case the time decay approach might have a positive influence. To confirm this intuition we performed additional experiments. In these experiments, for CC beginning with about a time-window size of 50 days we start to get a performance not as good as window size of 14 days but close to it (please also refer to Figure 4.2). Finally, using a considerably larger window size of 100 days we reach the effectiveness level of 14 days that uses no time decay. However, for larger time-window sizes, as early as 120 days, the system effectiveness starts to worsen again. For the cosine measure, we have similar observations but for different numbers of days (e.g., with a time-window size of 21 days, it starts to give a good performance and then suddenly worsens after that). The negative impact of larger window sizes can be attributed to the fact that when the window size is too wide some old events are

still able to sustain their influence on the system. Perhaps, what the time decay function does is ‘conceptually’ changing the size of an unnecessarily large window size to a more suitable smaller size for individual event cases. However, after a certain window size time decaying becomes ineffective.

In a real time environment, the use of large time-window sizes, depending on the resources, can be detrimental to the system performance in terms of efficiency. For example, in our data set 100 days means about 60,000 news stories is stored in the time-window and this many similarity value computations. The computational cost can be considerable if stories come with a little time delay [40].

5.3 Effect of Using Named Entities

In our study, we also experimented with the use of named entities. For this purpose, we created a named entity collection of 60,267 items [59]. In addition to this list we also used some intuitive ways of recognizing named entities by looking at the contexts of words that begin with capital letters. In the NED experiments we used three different similarity scores using document vectors based on:

1. words other than named entities
2. only named entities
3. all words

We also test the triangularization approach described in [34]. It finds the closest story of the newcomer in the time-window in terms of (1), in terms of (2), and obtains the similarity between these two stories using (3) and uses the third similarity score in decision-making.

In our experiments, among these four approaches the case that uses all words alone provided the best performance. The reason of our mediocre results with

named entities can be attributed to the possibility that our test collection topic stories are not conducive to the use of named entities. For example, [34] also have similar observations in their experiments. More experiments using different approaches are needed to come to a conclusion about the use of named entities in the Turkish TDT.

5.4 Chapter Summary

In this chapter, we first explored the influence of the other similarity functions on the new event detection. We also conducted experiments on the combinations each pair measures. The test results illustrated the cosine similarity function yields least C_{DET} cost in the detection. Using ‘or-combination’ of cosine and CC decreases this cost. The statistical results indicate that this decrease is not significant, however; this decrease is looking promising. Secondly, we examine the effect of the time penalty function, in which lower weight is assigned to old stories in the sliding window, on the detection. However, we seen that the use of such approach make the system’s performance worser. Lastly, we analyzed the effect of the named entities in the detection. We tested four different approach for the use of the entities. The experimental results indicated that the use of named entities does not contribute into performance of the system.

Chapter 6

NED and TT in Bilkent News Portal

Bilkent News Portal is a multi-source news portal. It has been active for more than a year and as of May 25, 2009 it contains more than 434,000 news articles. In this chapter, we present the resources, which are currently aggregated, and some changes occurred in the implementation of news portal that uses the NED and TT methods.

6.1 News Resources

Bilkent New Portal currently aggregates headlines from eight different news resources. These are as follows.

- CNNTürk
- Cumhuriyet
- Haber7
- Milliyet

- Radikal
- TRT
- Yenişafak
- Zaman

We observed from the portal that the total number of news published daily is about 2000. In Figure 6.1, the daily number of news are depicted for each news provider. Each column in the figure shows total number of the news in that day.

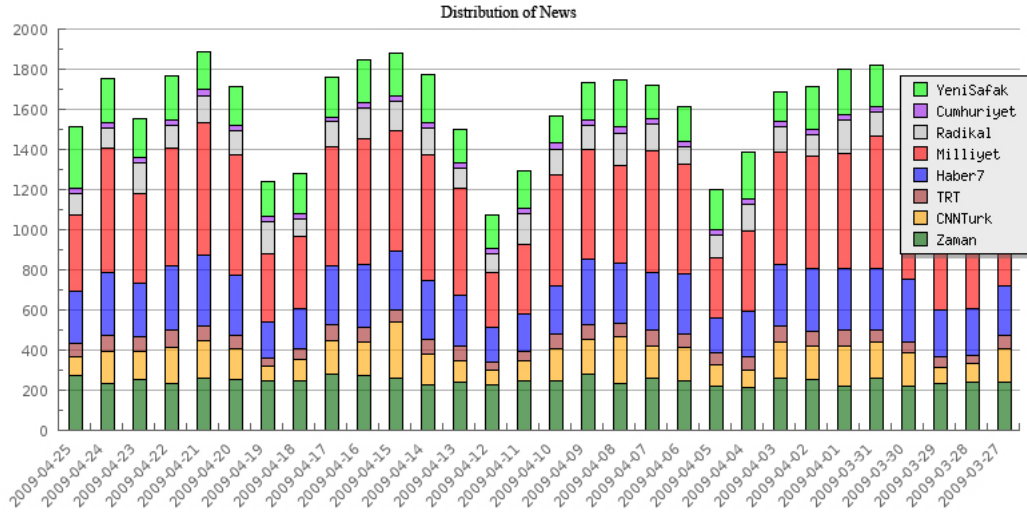


Figure 6.1: Daily number of news for each news resources.

One of the interesting result observed in Figure 6.1 is that the amount of news produced for each news provider does not vary a lot. These statistics are collected during one month (April 2009). It is observed that Milliyet news provider publishes the maximum amount of news for each day while Cumhuriyet produces the least amount. However, during handling this huge amount of news in the NED and tracking, we have encountered some problems, which are described in the next section.

6.2 Changes in NED and TT Methods for Bilkent News Portal Implementation

6.2.1 News Categorization Before NED and TT

We observed that our system found some stories as relevant to many topics even though they are not related, that is, high false alarm tracking occurs for these topics. In addition, these stories are also not in the same categorization of target topics. Therefore, we tested categorization methodology in order to improve the system effectiveness. In addition, we utilized the text categorization in NED. Fortunately, the current news resources provide us the information that each story belongs to which category. With use of this information, firstly, we classify all incoming stories into 6 categories. Then each incoming story is compared with the topics that belong to the same category. After applying this categorization methodology, we observed that the effectiveness of the tracking system has increased better.

6.2.2 Limiting Number of Topics for an Incoming Story

We limited a story to be relevant to n -number of topics and so each story is compared with the topics that belong to the same category. The story is restricted to be relevant to the topics that are chosen only highest n relevant topics. Normally, each topic is processed independent of the other topics, however; we ignore this so as to make the system yield fewer false alarms. Experimentally, we found the system increased the performance by means of this ignorance. The category information is unavailable in the BilCol2005 test collection; because of this the impact of categorization has not been formally evaluated. Note that during TT each topic has been considered separately and traditionally it is way the TT experiments are performed.

6.2.3 Using Named Entities During TT

We observed that some stories are so close to a topic that the tracking algorithm classified them as tracking events, even though they are actually not tracking stories. When we examined these kinds of stories, we found that they contain different named entities. For example, if we consider two car accidents, they are different topics according to the person names, the place name, and other named entities. Therefore, we utilized name entities in order to decrease false alarms in the tracking system. We modify the tracking system. The tracking system works normally. If an incoming story is found to be relevant with a topic, the similarity between the story and that topic is computed using only highest weighted n (experimentally use $n = 50$) names. If this similarity is above a pre-defined threshold, the story is labeled as relevant to that topic. With this methodology, we observed that the tracking system produces fewer false alarms.

Chapter 7

Conclusions

In this work, we study the TDT problem (more specifically NED and TT) in Turkish: it is a language domain that has never been investigated before, other than a small-scale study [36]. We tested the first large-scale Turkish TDT test collection containing 209,305 stories from five different web resources and 80 events annotated by 39 native speakers. We showed that in Turkish TDT, using CC (cover coefficient) and overlap similarity measures a simple word truncation stemming method can compete with a sophisticated stemming approach that pays attention to the morphological structure of the language. We illustrated that word stopping and the stopword list size are important for TDT as they affect system performance. We demonstrate that the decision-making confidence scores of two different similarity measures can be combined in a straightforward manner for improving system performance. In this study, we provide the first steps towards addressing the problem in a new language domain in this interesting and challenging research area. Our results provide benchmarks and baseline observations that can be used to gauge the research progress in this area, specifically in Turkish. Based on the results of our study, for similar TDT applications in the Turkish language we recommend using :

- The cosine similarity: when used with all document terms it provides the best performance, this recommendation is based on our experiments with

seven different similarity measures. System effectiveness with the cosine measure can be increased if its results and those of CC are combined for decision-making.

- Adaptive tracking: it provides a performance better than static tracking.
- A lemmatizer-based stemmer for indexing: it provides a better performance than those of no stemming and the fixed prefix (word truncation) method. If some decrease in system performance is tolerable and a quick implementation is needed then a simple word truncation method can be used. Indexing with no stemming is not a preferable option.
- A broad stopword list: the use of stopword list has an impact on performance and as its size increases effectiveness improves; however, the gain starts to diminish with the increasing list size.

Our study provides experimental results based on several different similarity measures when they are used alone or in pairs by using a simple and- or result combination approach. Other and more sophisticated result combination techniques, similarity measures or document representations can be used for improving the system effectiveness. Although we did not obtain good results in our preliminary experiments with named entities, we still believe that the use of named entities would be useful in the Turkish TDT. We speculate that named entities would have a positive influence in practical applications. Currently we work on reflecting our research results to a news portal [21, 46]. We continue our research on the use of named entities in the Turkish TDT.

There are several future research possibilities. For example, a system normally identifies several new events and most of them are not worth to bring to the attention of news consumers. It is important to identify and display new events that would attract common user interests. This needs further investigation. Furthermore, the concepts of the TDT research can be applied in various other data mining application areas that involve some sort of novelty detection, such as intelligence applications in detecting terrorist activities, identification of consumers' new spending trends, and email threading.

Bibliography

- [1] Milliyet news homepage. <http://www.milliyet.com.tr>, 2008.
- [2] Google news homepage. <http://news.google.com>, 2009.
- [3] Newsisfree: Your personal news portal. <http://www.newsisfree.com>, 2009.
- [4] Wikipedia homepage. <http://en.wikipedia.org/wiki/Portal:Currentevents>, 2009.
- [5] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, and U. Amherst. Topic Detection and Tracking Pilot Study. <http://ciir.cs.umass.edu/allan/Papers/bntuw98.ps>, 1998.
- [6] J. Allan, V. Lavrenko, and M. Connell. A Month to Topic Detection and Tracking in Hindi. *ACM Transactions on Asian Language Information Processing*, pp. 85–100, 2003.
- [7] J. Allan, V. Lavrenko, and H. Jin. First Story Detection in TDT Is Hard. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pp. 374–381, New York, NY, USA, 2000. ACM.
- [8] J. Allan, V. Lavrenko, and R. Swan. Explorations within Topic Tracking and Detection. In *Topic Detection and Tracking, Event-based Information Organization*, chapter 20, pp. 197–224. Kluwer Academic Publishers, 2002.
- [9] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR '98: Proceedings of the 21st annual international ACM*

- SIGIR conference on Research and development in information retrieval*, pp. 37–45, New York, NY, USA, 1998.
- [10] I. S. Altıngövd, E. Demir, F. Can, and Özgür Ulusoy. Incremental cluster-based retrieval using compressed cluster-skipping inverted files. *ACM Trans. Inf. Syst.*, 26(3):1–36, 2008.
- [11] K. Altıntaş, F. Can, and J. M. Patton. Language change quantification by time-separated parallel translations. *Literary & Linguistic Computing*, 22(4):375–393, 2007.
- [12] M. R. Anderberg. *Cluster Analysis for Applications*. Monographs and Textbooks on Probability and Mathematical Statistics. Academic Press, Inc., New York, 1973.
- [13] O. Bağlıoğlu. New Event Detection Using Chronological Term Ranking. Master’s thesis, Computer Engineering Department, Bilkent University, 2009.
- [14] G. Bennett, F. Scholer, and A. Uitdenbogerd. A comparative study of probabilistic and language models for information retrieval. In *ADC ’08: Proceedings of the nineteenth conference on Australasian database*, pp. 65–74, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.
- [15] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *SIGIR ’03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 330–337, New York, NY, USA, 2003.
- [16] J. P. Callan, W. B. Croft, and S. M. Harding. The Inquiry Retrieval System. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pp. 78–83. Springer-Verlag, 1992.
- [17] F. Can. Incremental clustering for dynamic information processing. *ACM Transactions on Information Systems*, 11(2):143–164, 1993.
- [18] F. Can, I. S. Altıngövd, and E. Demir. Efficiency and effectiveness of query processing in cluster-based retrieval. *Information Systems*, 29(8):697–717, 2004.

- [19] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar. Bilkent news portal: a personalizable system with new event detection and tracking capabilities. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 885–885, New York, NY, USA, 2008.
- [20] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar. Topic Detection and Tracking in Turkish (Under Revision). *Journal of the American Society for Information Science and Technology*, 2008.
- [21] F. Can, S. Kocberber, E. Balcik, C. Kaynak, and H. C. Ocalan. Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology*, 59(3):407–421, 2008.
- [22] F. Can and E. A. Ozkaran. Concepts and effectiveness of the cover coefficient based clustering methodology for text databases. *ACM Transactions on Database Systems*, 15(4):483–517, 1990.
- [23] J. Carbonell, Y. Yang, J. Lafferty, R. D. Brown, T. Pierce, and X. Liu. CMU report on TDT-2: Segmentation, detection and tracking. In *Proceedings of the DARPA Broadcast News Workshop*, pp. 117–120. Morgan Kaufmann Publishers, Inc., 1999.
- [24] CMU TEAM-A. Topic Detection and Tracking in 2004. www.nist.gov/speech/tests/tdt/2004/papers/CMU_A_tracking_TDT2004.ppt, 2004.
- [25] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan. UMass at TDT 2004. In *Working Notes of the TDT-2004 Evaluation*, 2004.
- [26] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison Wesley, 2008.
- [27] G. Cselle, K. Albrecht, and R. Wattenhofer. Buzztrack: topic detection and tracking in email. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pp. 190–197, New York, NY, USA, 2007. ACM.

- [28] F. Ekmekcioglu and P. Willett. Effectiveness of stemming for Turkish text retrieval. 34(2):195–200, 2000.
- [29] J. Fiscus and B. Wheatley. Overview of the TDT 2004 evaluation and results. <http://www.nist.gov/speech/tests/tdt/tdt2004/papers/NIST-TDT2004.ppt>, 2004.
- [30] J. G. Fiscus and G. R. Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking: event-based information organization*, pp. 17–31. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- [31] M. A. Hafer and S. F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385, 1974.
- [32] V. Hatzivassiloglou, L. Gravano, and A. Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 224–231, New York, NY, USA, 2000.
- [33] G. Kowalski and M. T. Maybury. *Information Storage and Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
- [34] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 297–304, New York, NY, USA, 2004.
- [35] G. Kumaran, J. Allan, and A. McCallum. Classification Models for New Event Detection. IR 362, University of Massachusetts, 2004.
- [36] H. Kurt. On-Line New Event Detection and Tracking in A Multi-Resource Environment. Master's thesis, Computer Engineering Department, Bilkent University, 2001.

- [37] T. Leek, R. Schwartz, and S. Sista. Probabilistic approaches to topic detection and tracking. In *Topic detection and tracking: event-based information organization*, pp. 67–83. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
- [38] D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 298–306, New York, NY, USA, 1996.
- [39] B. Li, W. Li, Q. Lu, and M. Wu. Profile-based event tracking. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 631–632, New York, NY, USA, 2005.
- [40] G. Luo, C. Tang, and P. S. Yu. Resource-adaptive real-time new event detection. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 497–508, New York, NY, USA, 2007.
- [41] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple Semantics in Topic Detection and Tracking. *Information Retrieval*, 7(3–4):347–368, 2004.
- [42] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings Eurospeech '97*, pp. 1895–1898, Rhodes, Greece, 1997.
- [43] K. R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman. Tracking and summarizing news on a daily basis with Columbia’s Newsblaster. In *Proceedings of the Human Language Technology Conference*, 2002.
- [44] NIST. The topic detection and tracking (TDT2002) task definition and evaluation plan (Technical Report Version 1.1). National Institute of Standards and Technology, 2002.
- [45] R. Nuray and F. Can. Automatic ranking of information retrieval systems using data fusion. *Information Processing Management*, 42(3):595–614, 2006.

- [46] H. C. Ocalan. Bilkent News Portal: A System With New Event Detection and Tracking Capabilities. Master's thesis, Computer Engineering Department, Bilkent University, 2009.
- [47] K. Oflazer. Two-level description of Turkish morphology. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, pp. 472–472, Morristown, NJ, USA, 1993.
- [48] R. Papka. *Online New Event Detection, Clustering, and Tracking*. PhD thesis, Computer Science Department, University of Massachusetts at Amherst, 1999.
- [49] R. Papka and J. Allan. Document classification using multiword features. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pp. 124–131, New York, NY, USA, 1998. ACM.
- [50] R. Papka and J. Allan. On-line New Event Detection using Single Pass Clustering. Technical report, 1998.
- [51] D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. Newsinessence: summarizing online news topics. *Communications of the ACM*, 48(10):95–98, 2005.
- [52] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [53] G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. In *Information Processing and Management*, pp. 513–523, 1988.
- [54] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. In *Readings in information retrieval*, pp. 355–364. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA, 1997.
- [55] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

- [56] R. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 215–223, New York, NY, USA, 1998.
- [57] A. D. Troy and G.-Q. Zhang. Enhancing relevance scoring with chronological term rank. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 599–606, New York, NY, USA, 2007.
- [58] N. Unwin. The individuation of events. *Mind*, 105(418):315–33, 1999.
- [59] E. Uyar. Near Duplicate News Detection Using Named Entities. Master’s thesis, Computer Engineering Department, Bilkent University, 2009.
- [60] C. J. van Rijsbergen. *Information Retrieval*. 2nd ed. London: Butterworths, 1979.
- [61] A. Vural. Online new event detection and Clustering Using the Concepts of the Cover Coefficient-based Clustering Methodology. Master’s thesis, Computer Engineering Department, Bilkent University, 2002.
- [62] C. L. Wayne. Topic detection and tracking overview and perspective. In *DARPA Broadcast News Transcription and Understanding Workshop*, p. 98, 1998.
- [63] J. P. Yamron, L. Gillick, S. Knecht, S. Lowe, and P. V. Mulbregt. Statistical models for tracking and detection. In *Proceedings of the DARPA Topic Detection and Tracking Workshop*, pp. 0885–6125, 2000.
- [64] J. P. Yamron, S. Knecht, and P. V. Mulbregt. Dragons tracking and detection systems for the TDT2000 evaluation. In *In Topic Detection and Tracking Workshop*, pp. 75–79. 2000.
- [65] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1999.

- [66] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 28–36, New York, NY, USA, 1998.
- [67] K. Zhang, J. Zi, and L. G. Wu. New event detection based on indexing-tree and named entity. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 215–222, New York, NY, USA, 2007.
- [68] W. Zheng, Y. Zhang, Y. Hong, J. Fan, and T. Liu. Topic Tracking Based on Keywords Dependency Profile. In *Information Retrieval Technology*, pp. 129–140, 2008.

Appendix A

Stopword Lists

Table A.1: Stopword list (10 words)

bir	çok	daha	için	olarak
bu	da	de	ile	ve

Table A.2: Stopword list (147 words) [20]

ama	böylece	eden	hiç	mi	olsun	tarafından
ancak	bu	ederek	hiçbir	mu	olup	üzere
arada	buna	edilecek	için	mü	olur	var
ayrıca	bundan	ediliyor	ile	nasıl	olursa	vardı
bana	bunlar	edilmesi	ilgili	ne	oluyor	ve
bazı	bunları	ediyor	ise	neden	ona	veya
belki	bunların	eğer	işte	nedenle	onlar	ya
ben	bunu	etmesi	itibaren	o	onları	yani
beni	bunun	etti	itibariyle	olan	onların	yapacak
benim	burada	ettiği	kadar	olarak	onu	yapılan
beri	çok	ettiğini	karşın	oldu	onun	yapılması
bile	çünkü	gibi	kendi	olduğu	oysa	yapıyor
bir	da	göre	kendilerine	olduğunu	öyle	yapmak
birçok	daha	halen	kendini	olduklarını	pek	yaptı
biri	de	hangi	kendisi	olmadı	rağmen	yaptığı
birkaç	değil	hatta	kendisine	olmadığı	sadece	yaptığını
biz	diğer	hem	kendisini	olmak	siz	yaptıkları
bize	diye	henüz	ki	olması	şey	yerine
bizi	dolayı	her	kim	olmayan	şöyle	yine
bizim	dolayısıyla	herhangi	kimse	olmaz	şu	yoksa
böyle	edecek	herkesin	mı	olsa	şunları	zaten

Table A.3: 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	öyleyi	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 B

Annotated Events

Table B.1: Summary information for annotated events

Event No.	Brief Description of Event(Event Type)	No. of Tracking Stories	Time Span (days)	Event Days Month/Day
1	Kars'ta Trafik Kazası 7 ölü (5)	20	203	5/28 - 12/16
2	Onur Air'in Hollanda'ya inişi yasaklandı (3)	159	203	5/12 - 11/30
3	Koreli bilim adamının kök hücre araştırması sahte (7)	8	11	12/19 - 12/29
4	Nema karşılığı kredi (8)	31	280	2/8 - 11/14
5	Tokyo'da trenlerde haremlik selamlık (13)	8	263	4/4 - 12/22
6	Londra metrosunda patlama (6)	454	175	7/7 - 12/28
7	Barbaros çocuk Köyü'nde çocuk tacizi skandalı (2)	88	275	1/26 - 10/27
8	Formula G (10)	20	58	7/4 - 8/30
9	Karamürsel kaymakamı intihar etti (2)	6	7	1/4 - 1/10
10	400 koyun intihar etti (5)	10	8	7/8 - 7/15
11	şemdinli olayları (2)	317	53	11/9 - 12/31
12	Türkiye'de kuş gribi (13)	229	83	10/10 - 12/31
13	şampiyon Fenerbahçe (10)	115	222	5/22 - 12/29
14	Mortgage Türkiye'de (9)	375	357	1/7 - 12/29
15	2005 Avrupa Basketbol şampiyonası (10)	78	297	1/15 - 11/7
16	Yüzüncü Yıl üniversitinde ihale yolsuzluğu iddiası (2)	326	79	10/14 - 12/31
17	Kral Fahd hastaneye kaldırıldı (12)	51	77	5/27 - 8/11
18	Memurlarının bir üst dereceye terfisi (9)	52	110	1/6 - 4/25
19	Bill Gates Türkiye'ye geldi (12)	17	8	1/30 - 2/6
20	Mısır'da patlamalarda çok sayıda kişi öldü (6)	120	43	7/23 - 9/3
21	Atill İlhan vefat etti (12)	40	70	10/11 - 12/19
22	Ata Türk'ün ölümü (12)	43	47	9/18 - 11/3
23	DT Genel Müdürü Lemi Bilgin görevden alındı (12)	63	109	8/19 - 12/5
24	Universiade 2005 Yaz Spor Oyunları (10)	248	289	3/4 - 12/17
25	Yahya Murat Demirel Bulgaristan'da yakalandı (3)	192	345	1/3 - 12/13
26	Bağdat El Ayma köprüsünde izdiham (6)	29	9	8/31 - 9/8

Table B.2: (Cont.) Summary information for annotated events

Event No.	Brief Description of Event(Event Type)	No. of Tracking Stories	Time Span (days)	Event Days Month/Day
27	Prof. Dr. Sadettin Güner'e saldırı (3)	41	291	1/8 - 10/25
28	Nestle'de mürekkepli süt (2)	11	2	11/22 - 11/23
29	Nermin Erbakan tedavi altında (12)	45	46	10/20 - 12/4
30	Ulubey'de çocukla annenin peş peşe ölümü (13)	6	31	5/19 - 6/18
31	15. Akdeniz Oyunları (10)	193	86	5/2 - 7/26
32	Kemal Derviş'in UNDP başkanı seçilmesi (8)	118	181	3/11 - 9/7
33	Irak başbakanı Caferi Tahran'ı ziyaret etti (11)	22	94	7/5 - 10/6
34	Gediz'de grizu patlaması (5)	39	36	4/21 - 5/26
35	Sarıgül'ün CHP'de kendini savunması (11)	110	352	1/2 - 12/19
36	Paris'de polisle göçmenler arasındaki çatışma (6)	245	51	10/29 - 12/18
37	Rock'n Coke açık hava müzik etkinliği (13)	11	5	9/2 - 9/6
38	Ankara Garı'nda tren kazası (5)	13	5	1/13 - 1/17
39	2005 Nobel tıp ödülü (7)	19	75	10/3 - 12/16
40	Kayseri Erciyes üniversindeki bebek ölümleri (2)	39	60	8/3 - 10/1
41	Marburg virüsünden ölenler (13)	25	65	3/16 - 5/19
42	Gamze Özçelik'in görüntüleri (2,12)	43	116	8/29 - 12/22
43	Türkiye'nin ilk yediz bebekleri geliyor (7)	56	301	2/17 - 12/14
44	Yeni Türk Ceza Kanunu (9)	53	193	6/1 - 12/10
45	Saddam Hüseyin'in yargılanmasına başlandı (3)	182	72	10/19 - 12/29
46	Beylikdüzü çöpte patlama (6)	17	5	11/18 - 11/22
47	Endonezya'nın Bali Adası'nda 4 bomba patladı (6)	15	4	10/1 - 10/4
48	Sahte rakı (3)	323	182	3/1 - 8/29
49	Hindistan'da üç saldırıda 66 kişi öldü (6)	21	5	10/29 - 11/2
50	Bülent Ersoy ve Deniz Baykal polemliği (12)	52	132	8/19 - 12/28
51	Tahran'da askeri uçak düştü (5)	9	2	12/6 - 12/7
52	Sochi seferinde Ufuk-1 gemisi yanmaya başladı (5)	20	3	8/25 - 8/27
53	İstanbul'da kanalizasyonda işçiler zehirlendi (5)	9	2	12/5 - 12/6
54	İKadınlara copla müdahale eden polisler (3)	104	297	3/6 - 12/27
55	Kuşadası'nda minibüsdeki patlama (3, 6)	50	4	7/16 - 7/19
56	Esenboğa Havalimanı iç hatlarda yangın (5)	18	36	11/14 - 12/19
57	Zeytinburnu'nda bir evde patlama (3, 6)	28	4	8/8 - 8/11
58	CityplaceMalatya çocuk yuvasında işkence (2, 3)	192	67	10/26 - 12/31
59	ABD denizaltısı ile Türk gemisi çarpıştı (5)	7	1	9/5 - 9/5
60	Prof. Dr. Kalaycı suikast sonucu öldürüldü (3)	44	23	11/11 - 12/3
61	İlk yüz nakli (7)	14	17	12/1 - 12/17
62	15 yeni üniversite kurulmasına ilişkin kanun (9)	59	50	11/12 - 12/31
63	Gaziantep tanker patlaması (5)	33	7	8/6 - 8/12
64	Hakkari'de bomba patladı (3, 6)	10	4	7/29 - 8/1
65	Erzurum çocuk yuvasında bebek ölümlü (2)	9	3	11/4 - 11/6
66	Kızım Koyuncu'nun ölümü (12)	30	129	6/25 - 10/31
67	Melih Kibar'ın ölümü (12)	16	120	4/7 - 8/4
68	Sarıkarnış şehitleri anıldı (13)	5	3	12/23 - 12/25
69	Endonezya'da yolcu uçağı düştü (5)	15	2	9/5 - 9/6
70	Şanlıurfa'da köprü inşaatı çöktü (5)	7	2	4/13 - 4/14
71	Japonya Osaka'da tren kazası (5)	29	4	4/25 - 4/28
72	Manken Tuğçe Kazaz hristiyan oldu (12)	11	76	9/22 - 12/6
73	Fotoğraf sanatçısı Mehmet Gülbiz öldürüldü (3)	14	127	2/4 - 6/10
74	Atina'daki Kara Harp Okulu'nda Türk bayrağı olayı (2)	55	71	4/16 - 6/25
75	Maslak'ta patlama (3, 6)	30	18	10/15 - 11/1
76	Didim'de denize uçak düştü (5)	13	2	7/19 - 7/20
77	Rum yolcu uçağı düştü (5)	106	115	8/14 - 12/6
78	İstiklal Caddesindeki ağaçlar kaldırıldı (13)	8	16	11/2 - 11/17
79	Zeytinburnu'nda gemi battı (5)	38	3	3/13 - 3/15
80	İngiltere'de Osmanlı kültürü hakkında sergi açıldı (13)	22	103	1/1 - 4/13
Avg.	-	73	92	-
Min.	-	5	1	-
Max.	-	454	357	-