

**PAST, PRESENT, AND FUTURE ON NEWS
STREAMS: DISCOVERING STORY CHAINS,
SELECTING PUBLIC FRONT-PAGES, AND
FILTERING MICROBLOGS FOR PREDICTING
PUBLIC REACTIONS TO NEWS**

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

By
Çağrı Toraman
September, 2017

PAST, PRESENT, AND FUTURE ON NEWS STREAMS: DISCOVERING
STORY CHAINS, SELECTING PUBLIC FRONT-PAGES, AND FILTERING
MICROBLOGS FOR PREDICTING PUBLIC REACTIONS TO NEWS

By Çağrı Toraman

September, 2017

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Fazlı Can(Advisor)

İ. Sengör Altingövde

Gönenç Ercan

Uğur Gündükbay

Özgür Ulusoy

Approved for the Graduate School of Engineering and Science:

Ezhan Karışan
Director of the Graduate School

ABSTRACT

PAST, PRESENT, AND FUTURE ON NEWS STREAMS: DISCOVERING STORY CHAINS, SELECTING PUBLIC FRONT-PAGES, AND FILTERING MICROBLOGS FOR PREDICTING PUBLIC REACTIONS TO NEWS

Çağrı Toraman

Ph.D. in Computer Engineering

Advisor: Fazlı Can

September, 2017

News streams have several research opportunities for the past, present, and future of events. The past hides relations among events and actors; the present reflects needs of news readers; and the future waits to be predicted. The thesis has three studies regarding these time periods: We discover news chains using zigzagged search in the past, select front-page of current news for the public, and filter microblogs for predicting future public reactions to events.

In the first part, given an input document, we develop a framework for discovering story chains in a text collection. A story chain is a set of related news articles that reveal how different events are connected. The framework has three complementary parts that i) scan the collection, ii) measure the similarity between chain-member candidates and the chain, and iii) measure similarity among news articles. For scanning, we apply a novel text-mining method that uses a zigzagged search that reinvestigates past documents based on the updated chain. We also utilize social networks of news actors to reveal connections among news articles. We conduct two user studies in terms of four effectiveness measures: relevance, coverage, coherence, and ability to disclose relations. The first user study compares several versions of the framework, by varying parameters, to set a guideline for use. The second compares the framework with 3 baselines. The results show that our method provides statistically significant improvement in effectiveness in 61% of pairwise comparisons, with medium or large effect size; in the remainder, none of the baselines significantly outperforms our method.

In the second part, we select news articles for public front pages using raw text, without any meta-attributes such as click counts. Front-page news selection is the task of finding important news articles in news aggregators. A novel algorithm is introduced by jointly considering the importance and diversity of selected news articles and the length of front pages. We estimate the importance of news, based on topic modelling, to provide the required diversity. Then, we select important documents from important topics using a priority-based method that helps in fitting news content into the length of the front page. A user study is conducted to measure effectiveness and diversity. Annotation results show that up to 7 of 10 news articles are important, and up to 9 of them are from different topics. Challenges in selecting public front-page news are addressed with an emphasis on future research.

In the third part, we filter microblog texts, specifically tweets, to news events for predicting future public reactions. Microblog environments like Twitter are increasingly becoming more important to leverage people's opinion on news events. We create a new collection, called BilPredict-2017 that includes events including terrorist attacks in Turkey from 2015 to 2017, and also Turkish tweets that are published during these events. We filter tweets by using important keywords, analyze them in terms of several features. Results show that there is a high correlation between time and frequency of tweets. Sentiment and spatial features also reflect the nature of events, thus all of these features can be utilized in predicting the future.

Keywords: Filtering, front-page, microblog, news actor, news chain, news selection, public reaction, text mining, topic modeling, zigzagged search.

ÖZET

HABER AKIŞLARINDA GEÇMİŞ, GÜNÜMÜZ VE GELECEK: HABER ZİNCİRLERİNİN KEŞFİ, ANASAYFALARIN HABER SEÇİMİ, HABERE KARŞI TOPLUMSAL TEPKİNİN TAHMİNİ İÇİN MİKROBLOG FİLTRELENMESİ

Çağrı Toraman

Bilgisayar Mühendisliği, Doktora

Tez Danışmanı: Fazlı Can

Eylül 2017

Haber akışlarında olayların geçmiş, şimdiki ve gelecek zamanı ile ilgili birçok araştırma imkanı bulunmaktadır. Geçmiş zaman olayların ve aktörlerin ilişkileri barındırmakta; şimdiki zaman haber okuyucularının ihtiyaçlarını yansıtmakta; gelecek zaman ise tahmin edilmeyi beklemektedir. Bu tez, bahsedilen üç zaman dilimiyle ilgili şu bölümlerden oluşmaktadır: Geçmişte zikzaklı arama yaparak haber zincirlerini keşfetmekte, günümüz haberlerinden genel amaçlı anasayfa oluşturmakta ve mikroblog yazılarını toplumsal olay tahmini için haberlere göre filtrelemekteyiz.

İlk bölümde, verilen bir haber yazısına göre bir koleksiyon içerisinde haber zincirlerini keşfeden bir çerçeve geliştirmekteyiz. Haber zinciri, farklı haber yazılarının bir araya gelmesiyle oluşmakta ve farklı olayların nasıl bir araya geldiğini ortaya çıkarmaktadır. Geliştirdiğimiz çerçeve yöntem birbirini tamamlayan şu üç bölümden oluşmaktadır. i) Koleksiyonun taranması, ii) zincir ile zincire eklenecek aday arasındaki benzerliğin hesaplanması ve iii) haber yazıları arasındaki benzerliğin hesaplanmasıdır. Tarama işlemi için, güncellenen zincire göre önceki dokümanları tekrar inceleyen zikzaklı arama yapan yeni bir metin madenciliği yöntemi uygulamaktayız. Haber yazıları arasındaki ilişkilerin ortaya çıkarılması için ise haber aktörlerinin sosyal ağından faydalanılmaktadır. Etkinliğin dört farklı yöntem—ilgi, kapsam, ahenk ve ilişkilerin keşfi—açısından değerlendirildiği iki kullanıcı araştırması yapmaktayız. İlk kullanıcı araştırması çerçeve yöntemin farklı versiyonlarını kıyaslayarak kullanıcılara bir rehber oluşturmaktadır. İkincisi ise çerçeve yöntemi üç altçizgi yöntem

ile kıyaslamaktadır. Sonuçlara göre yöntemimiz ikili kıyaslamaların %61'inde, orta ya da büyük etki boyutunda istatistiksel anlamda farklı olacak şekilde, etkinliğin iyileşmesini sağlamaktadır. Öteki kıyaslamalarda herhangi bir altçizgi yöntemi bizim yöntemimizi istatistiksel olarak geçememektedir.

İkinci bölümde, tıklama sayıları gibi meta-özellikler kullanmadan, sadece düz metin kullanarak haberler için toplumsal anasayfa seçmekteyiz. Anasayfa haber seçimi, haber toplayıcılarında önemli haberlerin bulunmasıdır. Anasayfaların uzunluğu ve seçilen haberlerin önemi ve çeşitliliği beraber düşünülerek yeni bir algoritma geliştirilmektedir. Haberlerin önemini, çeşitliliği de sağlamak amacıyla, konu başlığı modelleme yöntemiyle tahmin etmekteyiz. Önemli dokümanları daha sonra önemli konu başlıklarından, anasayfa uzunluğunu dolduracak şekilde öncelik-tabanlı bir method ile seçmekteyiz. Etkinliğin ve çeşitliliği bir kullanıcı araştırmasıyla ölçmekteyiz. Sonuçlara göre haber yazılarının 10 tanesinin en çok yedi tanesi önemli bulunmakta, dokuz tanesi ise farklı konu başlıklarından gözükmektedir. İleride yapılacak araştırmalara yol göstermesi için genel amaçlı anasayfa seçimindeki zorluklardan da bahsetmekteyiz.

Üçüncü bölümde ise haber olaylarına karşı ileride gerçekleşecek toplumsal tepkiyi tahmin etmekte kullanılabilecek filtreleme işlemi gerçekleştirmekteyiz. Twitter gibi mikroblog ortamları, toplumun görüşlerini ortaya çıkarmasıyla gün geçtikçe daha fazla önem kazanmaktadır. Terör olayları gibi 2015 ve 2017 yılları arasında gerçekleşmiş olayı ve bu olaylar sırasında atılan tweet'leri içeren BilPredict-2017 adında yeni bir toplumsal tepki veri setini geliştirmiş durumdayız. Önemli kelimelere göre tweet'leri filtrelemekte ve bunları çeşitli özelliklere göre analiz etmekteyiz. Sonuçlar, frekans, duygusallık, yer ve zaman özelliklerinin haber olaylarının doğasını yansıttıklarından dolayı gelecek tahmininde yararlanılabileceklerini göstermektedir.

Anahtar sözcükler: Anasayfa, filtreleme, haber aktörü, haber seçimi, haber zinciri, konu başlığı modeli, metin madenciliği, mikroblog, toplumsal tepki, zikzaklı arama.

Acknowledgement

For their endless love and support, I thank my mom and dad, Ülkü and Abdullah, my brother, Teoman, my lovely Huriye, my children Oscar and Tarçın, my friends Hasan, Emre, and Sermetcan.

For his great directions and vision, I thank my supervisor, Fazlı Can.

I thank the committee members, İ. Sengör Altingövde, Gönenç Ercan, Uğur Güdükbay, Özgür Ulusoy, and the former member Hakan Ferhatosmanoğlu, for their valuable reviews.

I thank all annotators for their kind support, Bilkent University Computer Engineering Department for their financial support on both my studies and travels, and TÜBİTAK (The Scientific and Technological Research Council of Turkey) who supported the works of the second, third, and fourth chapters under grant numbers 113E249, 111E030, and 215E169, respectively.

Contents

1	Introduction	1
2	The Past: Discovering Story Chains	3
2.1	Motivation	3
2.2	Aim	4
2.3	Contributions	5
2.4	Related Work	6
2.4.1	Simple Story Chains	6
2.4.2	Cluster-based Story Chains	6
2.4.3	Complex Story Chains	7
2.4.4	Other Studies Related to Discovery of Story Chains	8
2.4.5	Named Entity Recognition	8
2.5	A Framework for Story-Chain Discovery	9
2.5.1	Scanning the Collection	10
2.5.2	Similarity of Candidate with News Chain	13
2.5.3	Similarity Between News Articles	14
2.6	Evaluation	17
2.6.1	Setup	18
2.6.2	User Study 1: Varying Framework Parameters	25
2.6.3	User Study 2: Comparison with Baselines	34
2.7	Discussion	36

2.7.1	Practical Considerations	36
2.7.2	Complexity Analysis	40
3	The Present: Selecting Public Front-pages	43
3.1	Motivation	43
3.2	Aim	44
3.3	Contributions	46
3.4	Related Work	46
3.4.1	News Selection	46
3.4.2	Diversity in Document Selection	48
3.5	Front-page News Selection based on Topic Modelling	49
3.5.1	Finding Topics	50
3.5.2	Finding Document Importance	51
3.5.3	Finding Topic Importance	52
3.5.4	Priority-based News Selection using Document and Topic Im- portance	54
3.6	Evaluation	57
3.6.1	Setup	57
3.6.2	User Study	58
3.6.3	Results and Discussion	61
4	The Future: Filtering Microblogs for Predicting Public Reactions to News	63
4.1	Motivation	63
4.2	Aim	64
4.3	Contributions	65
4.4	Related Work	65
4.4.1	Event Detection	65
4.4.2	Prediction with Social Media	66
4.4.3	Prediction of Public Reactions	66

4.5	Our Filtering System	67
4.5.1	Preprocessing Tweets	67
4.5.2	Sentiment Analyzer	68
4.5.3	Filtering	69
4.6	Analysis	70
4.6.1	Dataset	70
4.6.2	Setup	71
4.6.3	Results and Discussion	72
5	Conclusion and Future Work	82
5.1	Conclusion	82
5.2	Future Work	83
	Bibliography	85
A	Details of Discovering Story Chains	98
A.1	Output Chains	98
A.2	User Study for Comparing with Baselines	98
B	Details of Selecting Public Front-pages	103
B.1	Main User Study	103
B.2	Additional User Study	103
C	Details of Filtering Microblogs	105
C.1	Collection	105

List of Figures

2.1	A sample story chain with five documents that tells a story that connects public protests in Ukraine with Russian independence activists in Crimea.	4
2.2	An illustration of the proposed framework that uses a sliding-time window with zigzagged search, and a social network of news actors. . .	10
2.3	Discovery of a sample story chain by using a sliding-time window with zigzagged search.	12
2.4	The framework algorithm for story chain–discovery.	17
2.5	A sample screen for the tasks screen of annotation program.	22
2.6	A sample screen from the annotation program.	23
2.7	Annotation results for Decision 1: Proper window length.	28
2.8	Annotation results for Decision 2: <i>all members</i> vs. <i>weighted members</i>	30
2.9	Annotation results for Decision 3: <i>All news actors</i> vs. <i>top 500 news actors</i>	31
2.10	Annotation results for Decision 4: Similarity methods.	32
2.11	Screenshots (<i>top</i> : front page, <i>down</i> : user interface for parameter selection) from Bilkent News Portal where our framework for story-chain discovery is applied.	40
3.1	Overview of our front-page news selection approach.	50

3.2	A sample document collection with 5 documents, 2 topics, and 5 unique words is given to demonstrate how to find document and topic importance.	54
3.3	An illustration of selecting news for two different front-page lengths, based on the same news collection given in Figure 3.2.	56
3.4	Pseudocode for our front-page news selection approach.	56
3.5	A sample screenshot from the annotation program.	59
4.1	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “the champions of the 2015 Turkish Super League is Galatasaray.”	73
4.2	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “the terror attack in Dağlica.”	75
4.4	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “Alanyaspor qualified to the Turkish Super League.”	77
4.6	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “Magazine programmer confuses Madonna.”	79
4.3	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “Aziz Sancar won the Nobel Prize in Chemistry 2015.”	80
4.5	The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “the 10 th Year Anthem is forbidden in Bolu.”	81
C.1	A sample instance from BilPredict-2017.	105

List of Tables

2.1	Main statistics after detecting named entities in our news collection.	18
2.2	The most frequently seen 10 people in our news collection.	19
2.3	The most frequently seen 10 organizations in our news collection. . .	20
2.4	The most frequently seen 10 locations in our news collection.	21
2.5	The design of User Study 1: Eight versions (A to H) of the framework algorithm.	26
2.6	The details of the Friedman test for Decision 1 with respect to Figure 2.7.	29
2.7	The details of the Wilcoxon test for Decision 2 with respect to Figure 2.8, where “d” is mean difference, “p”, “Z,” and “r” are scores of the Wilcoxon test.	31
2.8	The details of the Wilcoxon test for Decision 3 with respect to Figure 2.9.	32
2.9	The details of the Friedman test for Decision 4 with respect to Figure 2.10.	33
2.10	The design of User Study 2: Comparing our framework algorithm, <i>hZZ</i> : Hybrid and Zigzagged Search, with three baselines, <i>sTDT</i> : Simple TDT, <i>aTDT</i> : Adaptive TDT, <i>GN</i> : Google News.	35
2.11	The average scores of all annotators for <i>sTDT</i> : Simple TDT, <i>aTDT</i> : Adaptive TDT, <i>GN</i> : Google News, <i>hZZ</i> : Hybrid and Zigzagged. The method(s) with the highest score is marked as bold.	37

2.12	The details of the Friedman test with respect to Table 2.11.	38
2.13	Pairwise comparisons of the methods in Table 2.11.	39
3.1	Average, median, standard deviation, minimum, and maximum of <i>ann-imp</i> and <i>ann-div</i> scores are listed for the user study of 19 annotators.	61
4.1	The selected news events for the analysis in filtering.	71
A.1	The output of our framework algorithm, for the case <i>Ukrainian Riots</i> , to be compared with baseline methods in the second user study. . . .	99
A.2	The output of our framework algorithm, for the case <i>Trucks Going to Syria</i> , to be compared with baseline methods in the second user study.	99
A.3	The output of our framework algorithm, for the case <i>Allegations to Fenerbahçe</i> , to be compared with baseline methods in the second user study.	100
A.4	The details of the user study that we conduct to compare the success of our story-chain discovery method with baseline methods, in terms of <i>relevance</i> and <i>coverage</i>	101
A.5	The details of the user study that we conduct to compare the success of our story-chain discovery method with baseline methods, in terms of <i>coherence</i> and <i>ability to disclose relations</i>	102
B.1	The details of annotation results of our front-page news selection method.	104
C.1	Sample filtered tweets for the instance from 2015 given in Figure C.1.	106

Chapter 1

Introduction

News streams have hidden research challenges for the past, present, and future of events. The past hides relations among events and actors; the present reflects needs of news readers; and the future waits to be predicted. The thesis has three studies regarding these time periods: We discover news chains using zigzagged search in the past, select front-page of current news for the public, and filter microblogs to predict future public reactions to events.

In the first study, Chapter 2, we present a framework to discover story chains in a given text collection for an input document. A story chain is a set of related text documents, each with a different event. Discovering story chains reveals how events are connected and, thus, enables users to easily understand the big picture of events. In our case, a story chain is constructed for a given document in a news collection. Our framework has three complementary parts that (a) scan the collection, (b) measure the similarity between chain-member candidates and the chain, and (c) measure similarity among news articles by exploiting lexical features and news actors. We discover story chains with a novel approach, called zigzagged search, that uses a

sliding-time window that updates the current chain incrementally. Contributions of this part, among others, are that we develop a novel story-chain discovery framework that employs zigzagged search and news actors, and answer several research questions related to the framework.

In the second study, Chapter 3, we develop a novel approach for public news selection by using only raw text. While selecting the public front page, editors may select worthless news unintentionally, or even according to their own points of view. We present a novel approach that employs topic modelling to find diversified public front pages, while taking into consideration the importance of news within topics. Our method selects the most important news articles in the most important topics with a priority-based method for fitting to the length of the front page. We do not use meta-attributes, but leverage raw text. Contributions of this part, among others, are that we develop a novel algorithm to select public front-page news, and, to the best of our knowledge, this is the first study that examines public front-page news selection using only raw text.

In the third study, Chapter 4, we filter a microblog collection, specifically tweets, according to their relevance to news events in order to exploit these tweets for predicting public reactions to the same events. Microblog environments like Twitter are increasingly becoming more important to leverage people’s opinion on news events. Given a news article as input, we fetch at most 5 days of tweets after the origin date of news event, and then preprocess tweets, which includes cleaning, normalization, and stemming steps. Filtered tweets are analyzed in terms of frequency, sentiment, temporal, and spatial features. Contributions of this part are that we filter microblog texts with a comprehensive analysis of several features to be used in predicting public reactions, and create a public-reaction dataset including terrorist attacks between 2015 and 2017.

In Chapter 5, we sum up this thesis with a brief conclusion and future work.

Chapter 2

The Past: Discovering Story Chains¹

2.1 Motivation

A story chain is a set of related text documents, each with a different event. In our case, a story chain is constructed for a given document in a news collection. We use the phrases “story chain” and “news chain” interchangeably. Discovering news chains (a) reveals how events are connected and, thus, enables users to easily understand the big picture of events, (b) makes news consumers become aware of hidden relations among events, (c) detects different aspects of the input story, and (d) helps avoid information overload. Some of the possible application domains of story-chain discovery are investigative journalism, in which journalists or researchers examine a specific news topic; the analysis of intelligence reports [2], patents [3], and legal documents [4].

¹This study is published in [1].

2.2 Aim

A good story chain has a set of properties. Shahaf and Guestrin [5] argue that relevance between input and chain members should be high. Coherence is another important property of news chains, which means a low relevance gap in the transition between any two chain members. Zhu and Oates [6] expand the characteristics of news chains with measures of low redundancy and high coverage. A story chain has low redundancy when it includes no more than one representative for each event, and high coverage when it covers different aspects of the story. In addition to these, we also consider if previously unknown relations among news actors are disclosed by the chain—we call it the ability to disclose relations. Figure 2.1 shows a sample story chain with five documents.

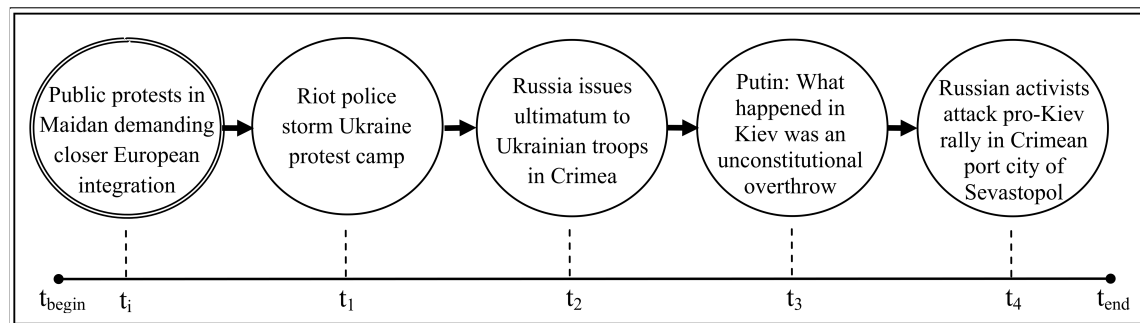


Figure 2.1: A sample story chain with five documents that tells a story that connects public protests in Ukraine with Russian independence activists in Crimea. The input document is double circled with timestamp t_i . The beginning and end of the collection are t_{begin} and t_{end} , respectively.

We present a framework for story-chain discovery based on three complementary parts that (a) scan the collection, (b) measure the similarity between chain-member candidates and the chain, and (c) measure similarity among news articles by exploiting lexical features and news actors. We discover story chains with a novel approach that uses a sliding-time window that updates the current chain incrementally. Inside the window, for the first time in this domain, we introduce a zigzagged search

that reinvestigates past documents based on the updated chain. Zigzagged search imitates the forward-and-backward search behavior of an investigative journalist.

In the next section, we list our contributions. We then summarize the related work for story-chain discovery, explain the details of our framework, present the user studies and their results, and finally conclude this chapter with some practical considerations and complexity analysis of our framework algorithm.

2.3 Contributions

The contributions of this chapter are the following. We

1. develop a story-chain discovery framework that employs zigzagged search and news actors,
2. conduct two user studies:
 - (a) The first finds a guideline for using the framework by answering the following research questions:
 - i. What is the proper time-window length to be used while scanning the collection?
 - ii. How should we measure the similarity between a news chain and a candidate article?
 - iii. When a social network of news actors is utilized, is it necessary to use a large network of news actors instead of exploiting a subset of important actors?
 - iv. Which similarity method performs better in news-chain discovery: lexical features using the vector space model, or meta features based

on news actors? Can we improve the effectiveness by using multiple methods together in a hybrid approach?

- (b) The second compares our method with baselines to answer:
 - i. What are the benefits of our framework against baseline approaches?
- 3. support user studies with statistical tests, which can set an example for similar studies,
- 4. and, integrate our framework into a real-time news aggregator to observe its practical implications.

2.4 Related Work

2.4.1 Simple Story Chains

In TDT (Topic Detection and Tracking) [7, 8], a topic is defined as an event or activity, with all directly related events. Since news articles are related to the same topic, we refer to such chains as simple story chains. TDT has a task called link detection that “detects whether a pair of stories discuss the same topic”. In our case, the purpose of story chains is not restricted to detect relations in the same topic, but also coherent connections among different topics.

2.4.2 Cluster-based Story Chains

Mei and Zhai [9] cluster similar documents to obtain trends or themes in time, and then, clusters are connected to exhibit evolutionary theme patterns. Subasic and

Berendt [10] examine evolutionary theme patterns using interactive graphs. Nallapati, Feng, Peng, and Allan [11] introduce event threading in clusters of events, and find dependencies among these clusters in a tree structure. Yan, Shi, and Wei [12] develop event evolution graphs, which present underlying structure and relations among events of a topic. Kim and Oh [13] apply topic modeling to uncover groups that contain related documents; chains are then constructed by finding similar topics in a time window. Song et al. [14] develop a topic modeling approach to model documents and concept drifts in a tree structure. Shahaf, Guestrin, Horvitz, and Leskovec [15] connect sets of clusters of news articles in a timeline to cover different aspects of the same topic. They find overlaps among clusters of different chains to reveal the evolution of the story.

2.4.3 Complex Story Chains

We define that a story chain is complex if it reveals relations among events of different topics. Complex story chains are first studied by Shahaf and Guestrin [5]. Given two input news articles, their aim is to find a coherent story chain that connects them by maximizing the influence of the weakest connection. Influence is a measure to find similarity between two documents using the random walk theory. Zhu and Oates [6] claim to improve the approach of Shahaf and Guestrin [5] in terms of efficiency and redundancy. They use the inner structure of news articles by extracting named entities. The main differences of our study from others are the following: (a) We introduce zigzagged search to discover story chains. (b) We exploit social networks of news actors to reveal connections among news articles. (c) Our user studies are supported by statistical tests. (d) We integrate our method into a real-time news aggregator to observe practical issues. (e) Our input is only a news article that indicates the start of a chain, instead of taking both start and end, i.e. connecting

two dots [5] (they do not develop an algorithm from scratch, but adapt their two-input algorithm to the one-input problem by extending it with user’s feedback).

2.4.4 Other Studies Related to Discovery of Story Chains

Timeline summarization methods give a summary of a query event with timeline projection [16]. Some studies create a hierarchy in timeline for the given text collection [17]. Giving two objects, storytelling studies [18] aim to explicitly relate them by using their intersections; for instance, two documents (abstracts) are linked by finding word intersections. Choudhary, Mehta, Bagchi, and Balakrishnan [19] find actors and their interactions in a given news collection. Similar studies support intelligence analysts to suggest unknown relations among entities [2].

2.4.5 Named Entity Recognition

There are some popular software tools for Named Entity Recognition (NER). GATE (General Architecture for Text Engineering) [20] is an open-source text analysis Java software developed by The University of Sheffield. It has a pipeline of NLP modules to extract information from plain text such as sentence splitter, tokenizer, POS (part of speech) tagger, and NER. Each of these modules has a language resource like tokenizer rules or sentence segmentation heuristics. Another tool is Stanford’s NER based on linear chain conditional random field (CRF) sequence models [21]. CRF is a hybrid machine-learning approach taking advantages of Hidden Markov Models and MaxEnt Markov Models. This approach is based on features such as previous/next words and prefixes/suffixes. Illinois NE Tagger [22] is another NER tool that employs several machine-learning algorithms such as Hidden Markov Models and Neural Networks. Readers are encouraged to examine Nadeau and Sekine’s

survey [23] for other open-source NER tools.

Although there are several NER tools, most of them do not support Turkish. An exception is JRC-Names [24] developed by EU Joint Research Center that supports multi-languages including Turkish. Since Turkish has an agglutinative morphology, using statistical models for Turkish NER results with the data sparseness problem. Modeling with morphological analysis for Turkish NER improves success in terms of F-Measure [25], [26]. Chain conditional random field (CRF) is also applied for Turkish NER by ITU Turkish NLP Group and results are promising [27]. Lastly, a recent study by Küçük and Yazıcı develops a comprehensive rule-based approach for Turkish NER that utilizes from several lexicon resources and pattern rules [28]. Lexical resources are used for creating list of people, locations etc. There are approximately 12,800 lexical resources and 260 patterns. Morphological analysis is used as well.

Since we observe, during our preliminary experiments, that the named-entity-recognition program of Küçük and Yazıcı detects more entities than that of both JRC-Names and ITU Turkish NLP Group’s CRF-based approach, we decide to use it in this study.

2.5 A Framework for Story-Chain Discovery

In this section, we present a temporal text-mining framework for story-chain discovery. Figure 2.2 shows an illustration of the framework that includes three complementary parts: (1) A given collection is scanned by using a sliding-time window that uses zigzagged search. Assume that the current chain includes four documents, labeled w , x , y , z ; the first candidate document to be added to the chain is labeled as a . (2) Documents are added to the chain according to the similarity between

the candidate and the chain. (3) Similarity between two documents is measured by employing a social network of news actors to reveal connections among news articles. We also calculate similarities based on the vector space, named-entity, and hybrid model.

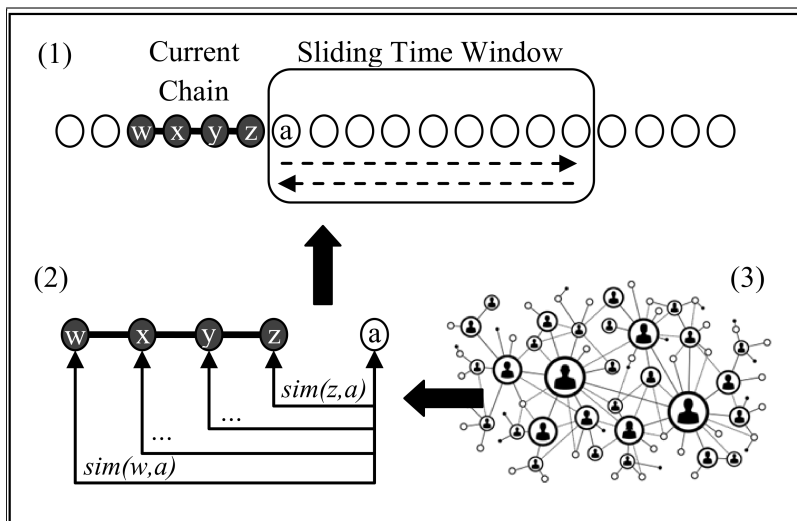


Figure 2.2: An illustration of the proposed framework that uses a sliding-time window with zigzagged search, and a social network of news actors.

2.5.1 Scanning the Collection

We scan a given collection to search news articles, related to the input, by using a sliding-time window that uses zigzagged search; Figure 2.3 shows an example. The timeline is divided into non-overlapping windows with a fixed-length (w) in days. We use a time window-based approach to update the current chain incrementally by considering only the members of the window. The user selects a news article d_i with the timestamp t_i , where $1 \leq i \leq N$, and N is the number of documents in the collection. An initial news chain is created with d_i . The first window is defined for $[t_i, t_i + w)$. The time window is not allowed to exceed the ending time of the collection. If the similarity between d_i and d_c , a candidate news article inside the window, is higher

than a threshold value , then d_c is added to the news chain.

We propose a forward-and-backward zigzagged search. We expect that making a zigzag in the timeline reveals missed news articles by using newly added documents to the chain. To do so, after processing the last news article in terms of time in the window, a new search phase on the same window is started by going backwards in the timeline. All news articles until the beginning of the window are processed in this backward-search phase. However, in this phase, similarity is calculated between the current chain, which is updated in the forward phase, and the candidate document. After a zigzag is completed, the same process is repeated by sliding the window by w days, until the remaining news articles are processed.

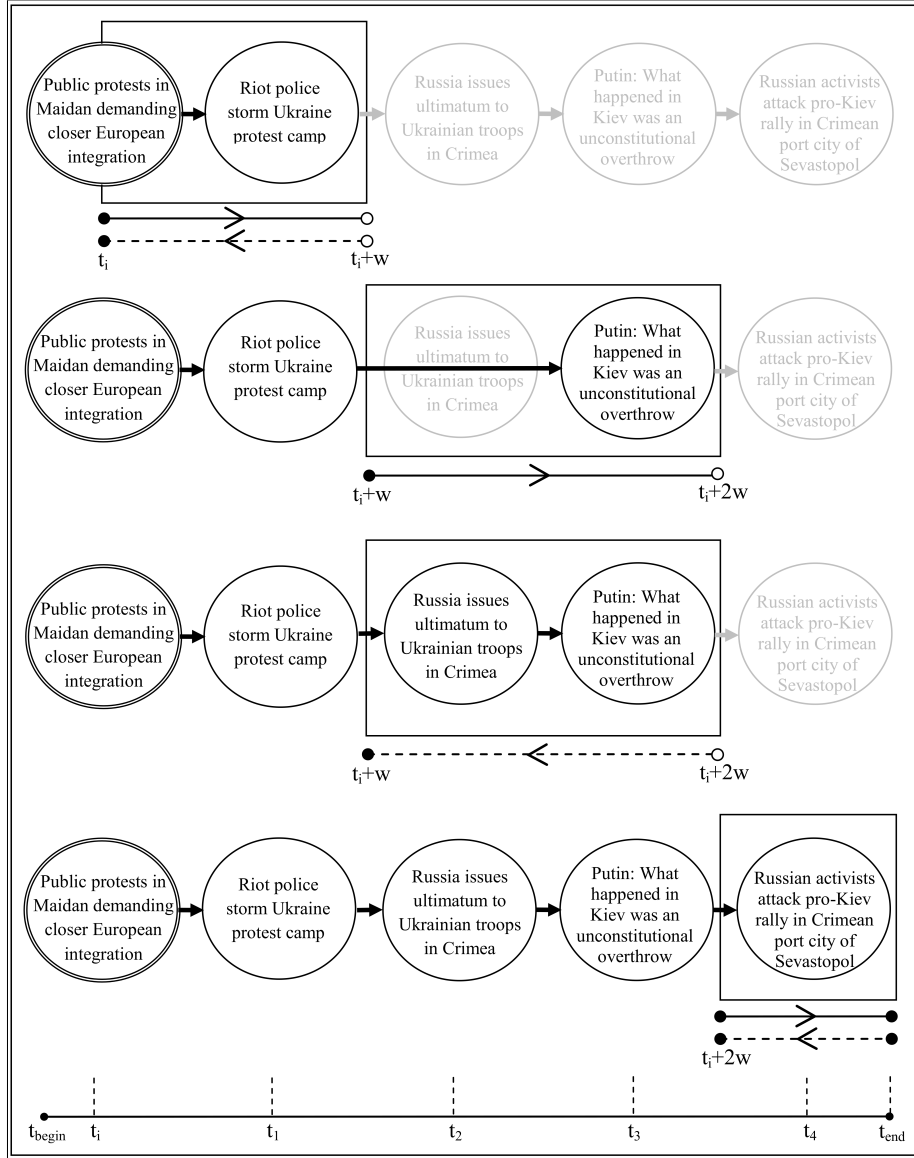


Figure 2.3: Discovery of a sample story chain by using a sliding-time window with zigzagged search. The beginning and end of the collection is t_{begin} and t_{end} , respectively. Window length is w days. The beginning of a window is inclusive, and the end is exclusive, shown by filled-in and empty circles, respectively. The input is the double-circled news article that has the timestamp t_i , mentioning the beginning of public protests in Ukraine in 2014. After three windows are processed from t_i to t_{end} , the bottommost chain is the output chain with five documents telling a story that connects public protests in Ukraine with Russian independence activists in Crimea. This chain is an extracted version of the output of *hZZ*—namely, the hybrid algorithm, to be defined later, of the *Ukrainian Riots* case.

2.5.2 Similarity of Candidate with News Chain

While processing each candidate document to be added to the chain, we measure its similarity with the chain, which is represented by all of its current members. We also assign weights to similarity scores between the candidate and chain members. We call these methods all members and weighted members, respectively.

2.5.2.1 All members

Similarity scores between a candidate document, d_c , and chain members are measured as follows, where h is the current chain.

$$sim_{all}(d_c, h) = \frac{\sum_{i \in h} sim(d_i, d_c)}{|h|} \quad (2.1)$$

2.5.2.2 Weighted members

We assign weights to $sim(d_c, h)$, according to the closeness of the candidate document to the chain as follows, where $w_i = r_i/|h|$, and r_i is the order of the document, d_i , in the chain. We expect to improve the coherence of the chain. For simplicity, it is assumed that w_i is calculated to add a candidate document to the end of the chain.

$$sim_{weighted}(d_c, h) = \frac{\sum_{i \in h} sim(d_i, d_c) \times w_i}{|h|} \quad (2.2)$$

2.5.3 Similarity Between News Articles

We propose four methods for measuring the similarity between two documents.

2.5.3.1 Vector space–based similarity

In the vector space model, documents are represented with word vectors that are sets of unique tokens in the collection. Each word is assigned to a weight by using term frequency. We calculate similarity between two document vectors by the cosine similarity measure. We use a stop word list—an extended version of the list given in [29]—and *F5* stemming, which uses the first five letters of each word, and shows good performance in information retrieval [29] and news categorization [30]. We use the phrases vector space model and cosine similarity interchangeably.

2.5.3.2 Named entity–based similarity

Named entity recognition (NER) is the task of information extraction to identify and classify important elements in a text document [23]. In this study, named entities are detected for people, organizations, and locations. We employ the named-entity-recognition program of Küçük and Yazıcı [28].

The output of Küçük and Yazıcı’s algorithm is too noisy, since there are lots of first name that are recognized without its last name, and several named entities refer to the same meaning, which can be solved by named entity resolution [31]. Named entity resolution is a difficult task for Turkish, therefore we manually resolve named entities that refer to the same object. For instance, Atatürk and Gazi Mustafa Kemal are two named entities referring to the same person. Manual named entity resolution is done as follows: First, all named entities labeled by Küçük and Yazıcı’s algorithm

are obtained. Then, we write heuristic rules for named entities if they can be resolved by their previous or next tokens. Also, we merge synonyms of popular objects into the same named entity.

After named entities are determined, the similarity between two news articles, d_i and d_j , is measured by the Dice similarity coefficient, as follows, where N_c is the number of common unique actors in d_i and d_j ; N_i and N_j are the number of unique actors in d_i and d_j , respectively.

$$sim_{named-entity}(d_i, d_j) = \frac{2 \times (N_c)}{N_i + N_j} \quad (2.3)$$

2.5.3.3 Social network-based similarity

Social network studies aim to reveal relations among social actors in a network structure [32]. We create a social network of news actors (named entities) for the entire collection, where edges represent relations. We detect news actors as described in above, and create an edge between two actors if both occur in the same document. We use the Dice coefficient for assigning weights to edges. The edge weight, $w(a, b)$, between two actors, a and b , in a social network is determined as follows, where N_c is the number of documents in which both actors occur, N_a and N_b are the numbers of documents that include the actors a and b .

$$w(a, b) = \frac{2 \times (N_c)}{N_a + N_b} \quad (2.4)$$

The similarity between two documents, d_i and d_j , is then measured as follows, where A_i and A_j are the sets of unique actors in d_i and d_j , and N_p is the number of all unique pairs between the actors of d_i and d_j .

$$sim_{social-network}(d_i, d_j) = \left(\sum_{a \in A_i} \sum_{b \in A_j} w(a, b) \right) / N_p \quad (2.5)$$

The difference between the named entity- and social network-based similarity methods is that the former considers only the co-occurrence of actors between two news articles; the latter uses edge weights in a social network, i.e., relations among actors of two news articles.

2.5.3.4 Hybrid similarity

The hybrid similarity between two documents, d_i and d_j , is a linear combination of the similarity scores of n methods:

$$sim_{hybrid}(d_i, d_j) = \sum_{k=1}^n sim_k(d_i, d_j) \times \alpha_k \quad (2.6)$$

Each method k outputs a score for the similarity between d_i and d_j as $sim_k(d_i, d_j)$; however, there is a need for the calibration of different methods. The parameter α_k is a significance coefficient for the method k ($0 \leq \alpha_k \leq 1$, $\sum_{k=1}^n \alpha_k = 1$). We combine lexical features, namely the vector space model, and social network, in the hybrid model by setting α values equal to 0.5.

The framework algorithm for story chain-discovery is given in Figure 2.4.

Input: News collection D in temporal order, input news article d_i , social network of news actors SN , window length w , similarity thresholds θ_{vsm} and θ_{sn} , hybrid weights α_{vsm} and α_{sn} .

Output: News chain C that tells a story about d_i .

```

1   $t_j \leftarrow t_i$ , timestamp of  $d_i$  // start of the first window
2   $C \leftarrow \{d_i\}$  // construct the initial chain
3   $D_s \leftarrow$  subset of  $D$  after  $t_j$  to the end of  $D$ 
4   $D' \leftarrow D_s$ 
5  while  $D'$  is not empty do
6      Create window  $W_j$  for  $[t_j, t_j + w)$  // cannot exceed the end of the collection
7      Create document set  $D_j$  for  $W_j$ 
8      for  $\forall d_k \in D_j$  do // the zig (forward) phase
9          Find the similarity between  $d_k$  and  $C$ ,  $sim(d_k, C)$ , using  $SN$  if necessary
10         if  $sim(d_k, C) > \theta$  then
11             // use  $\theta_{vsm}$ ,  $\theta_{sn}$ , and their weighted average with respect to  $\alpha_{vsm}$  and  $\alpha_{sn}$ 
12             // for the vector space-, social network-based, and hybrid methods, respectively
13              $C \leftarrow C \cup d_k$ 
14         end-if
15     end-for
16     Reinvestigate missed documents in reverse temporal order by repeating lines 8-13 using the
17     updated chain // the zag (backward) phase
18      $D' \leftarrow (D_s \setminus D_j)$ 
19      $t_j \leftarrow t_j + w$ 
20 end-while

```

Figure 2.4: The framework algorithm for story chain-discovery.

2.6 Evaluation

To the best of our knowledge, there is no ground truth for the evaluation of story-chain discovery algorithms. For this reason, we conduct (two) user studies². The first compares several versions of the framework, by varying parameters, to set a guideline for use. The second compares the framework with three baseline methods.

²The materials that we are unable to give due to the limited space are provided in the details web page (<https://github.com/BilkentInformationRetrievalGroup/TUBITAK113E249>); such as the text collection, output story chains, annotations, and details of statistical tests. Output chains and annotation scores of the second user study are also given in Appendix A.

2.6.1 Setup

2.6.1.1 News Collection

Chains are discovered in a news collection that includes 1,656 documents from the *Sözcü* newspaper (<http://www.sozcu.com.tr>) between December 20, 2013 and March 11, 2014.

Structure of a news article in the collection consists of the following seven tags.

1. *DOCNO*. Unique document identifier that is composed of seven digits.
2. *SOURCE*. Name of the RSS feeder.
3. *URL*. URL address that publishes news article online.
4. *DATE*. Publication date of news article in the format of year/month/day.
5. *TIME*. Publication time of news article in the format of hour:minute:second.
6. *HEADLINE*. Headline of news article.
7. *TEXT*. Body text of news article.

The number of detected named entities are given in Table 2.1.

Table 2.1: Main statistics after detecting named entities in our news collection.

<i>Type</i>	<i>Count</i>
Documents	1656
Unique people	2890
Unique organizations	915
Unique locations	1152

Tables 2.2, 2.3, and 2.4 list the most frequently seen 10 people, organizations, and locations respectively in our news collection. Although we manually resolve named entities, there are still some unresolved named entities like TOPBAŞ and DENİZ in Table 2.2. There are several people whose last names are TOPBAŞ. We can resolve full names by looking their previous and next tokens; but previous and next tokens are useless in 33 cases. Another problem is seen with DENİZ. The tool that we use for named-entity detection mostly labels DENİZ as a person name; but there are several objects including DENİZ like DENİZ PİYADE or DENİZ FENERİ. We need more advanced algorithms for named entity resolution in such cases.

Table 2.2: The most frequently seen 10 people in our news collection.

<i>Person</i>	<i>Number of Documents</i>
RECEP TAYYİP ERDOĞAN	363
BİLAL ERDOĞAN	79
FETHULLAH GÜLEN	57
ABDULLAH GÜL	55
DEVLET BAHÇELİ	41
MUSTAFA SARIGÜL	40
MUSTAFA KEMAL ATATÜRK	36
KEMAL KILIÇDAROĞLU	35
TOPBAŞ	33
DENİZ	32

Frequency distributions of people, organizations, and locations are tested whether they fit into power-law distribution. We test if data follows power-law distribution with a goodness-of-fit test [33]. This test is based on a hypothesis that says data is generated from a power-law distribution and outputs a p -value that can be used to quantify the validity of hypothesis. If the p -value is close to 0 (more specifically smaller than 0.1), then hypothesis is rejected, which means data is not fit into a power-law distribution. If it is higher than 0.1, then data is plausible for fitting a power-law distribution. For frequency distributions of people, organizations, and

Table 2.3: The most frequently seen 10 organizations in our news collection.

<i>Organization</i>	<i>Number of Documents</i>
AKP	294
CHP	234
TBMM	182
MHP	91
ADALET	76
AB	65
MİT	54
BAŞBAKANLIK	50
EMNİYET MÜDÜRLÜĞÜ	50
HÜRRIYET	46

locations, p -values are obtained as 0.56, 0.95, 0.12 respectively. Thus, we can conclude that frequency distributions of people and organizations are plausible to fit into power-law while frequency distribution of locations is barely plausible.

Three news cases (topics) are used as input in our user studies—Shahaf and Guestrin, and Zhu and Oates manually select 5 and 3 cases, respectively. The first case is the riots and protests against the Ukrainian government, demanding closer European integration, which started in November 2013, and referred to as *Ukrainian Riots* in this study. The second case is the trucks that were pulled over while going from Turkey to Syria by military police, claiming that they carry illegal ammunition, in January 2014, referred to as *Trucks Going to Syria*. The last case is the domestic match-fixing allegations to the Fenerbahçe football team, started in July 2011, referred to as *Allegations to Fenerbahçe*. We select three input news articles representing the cases. The dates of the input documents for each topic are January 25, 2, and 17 of 2014, respectively.

Table 2.4: The most frequently seen 10 locations in our news collection.

<i>Location</i>	<i>Number of Documents</i>
TÜRKİYE	435
İSTANBUL	360
ANKARA	176
ABD	122
İZMİR	105
SURİYE	71
AVRUPA	68
AMERİKA	61
RUSYA	53
AYDIN	50

2.6.1.2 Annotation Program

User studies are conducted on an annotation program written in Java. Annotators are assigned to the same tasks.

Annotation program consists of three different screens that are login, tasks and annotation screen. In order to continue labeling from the same state of annotation whenever annotators would like to do so, we assign each annotator a user name and password, which are asked in login screen. In the user study, each annotator is assigned to the same tasks. Annotators have to complete all news-chain-annotation tasks. A sample screen-shot for tasks screen is given in Figure 2.5. In tasks screen, annotators can see which tasks they have completed and how much time a particular task have taken. Annotators can also redo the annotation they have finished earlier.

News C...	Status	Start Date [y/m/d h:m:s]	End Date [y/m/d h:m:s]	Total Spent Time
1	COMPLETED	2015/11/07 11:40:01	2015/11/07 11:41:00	0 mins, 59 secs
2	COMPLETED	2015/11/06 12:21:41	2015/11/06 12:34:03	12 mins, 22 secs
3	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
4	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
5	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
6	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
7	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
8	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
9	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
10	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
11	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
12	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
13	NOT COMPLETED	N/A	N/A	0 mins, 0 secs
14	NOT COMPLETED	N/A	N/A	0 mins, 0 secs

Figure 2.5: A sample screen for the tasks screen of annotation program.

Annotators labels a particular news chain in annotation screen. A sample screenshot for annotation screen is given in Figure 2.6. Annotation screen consists of two main panels. At the top panel, news articles in the selected news chain are listed chronologically. Each news article is given with its date and snippet of first 200 characters. Full text is visible in a pop-up window if news article is double-clicked. Input news article's snippet is always bold and the sign of three consecutive stars is placed to its beginning to discriminate it from others. At the bottom of news articles, 6 questions are asked to annotators to assess the performance of given news chain. In this screen, annotation task is not completed unless all news articles are double-clicked and viewed in full text in order.

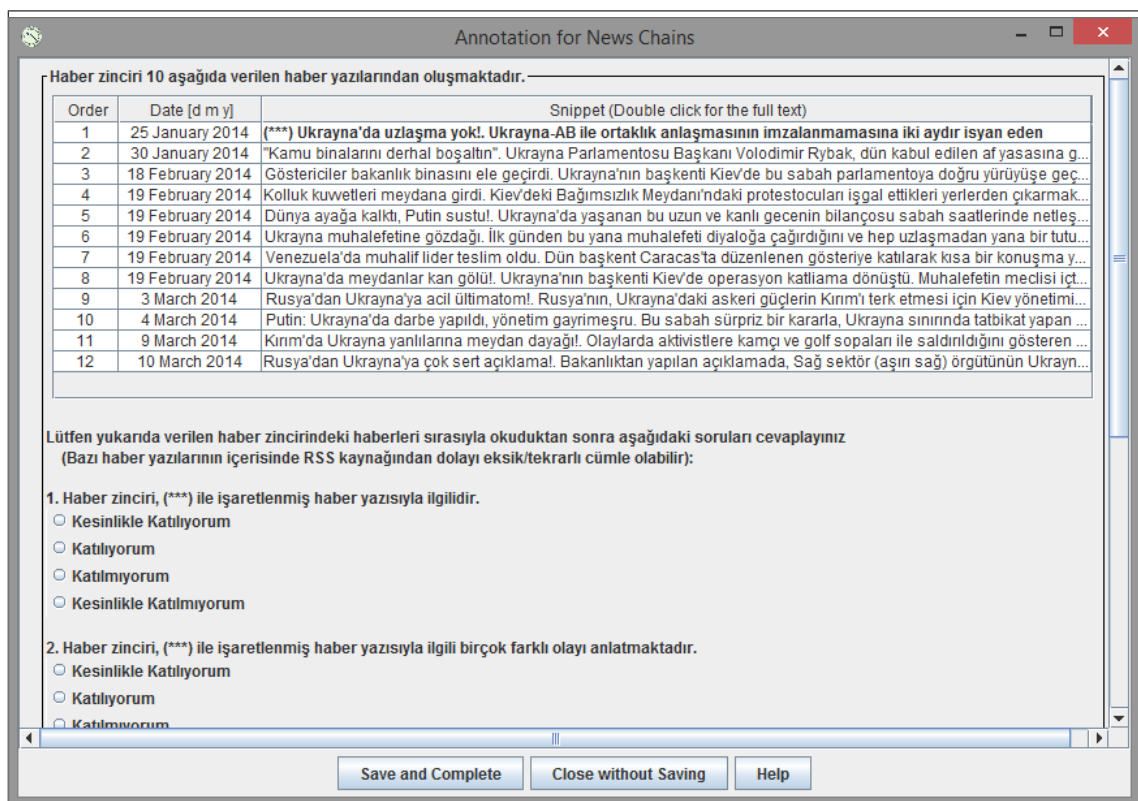


Figure 2.6: A sample screen from the annotation program. The current chain to be annotated, regarding *Ukrainian Riots*, is given at the top of screen. Annotators have to read all news articles in order, and then answer all questions.

2.6.1.3 Evaluation Measures

In similar studies, Shahaf and Guestrin [5] evaluate story chains according to relevance, coherence, and redundancy. Zhu and Oates [6] consider coverage, in addition to other measures. We also assess if previously unknown relations among news actors are disclosed by the chain—ability to disclose relations.

We give annotators five statements, and ask them to label to what extent they agree that (1) the news article is relevant to the input document marked with (***);

(2) the news chain covers different events related to the input; (3) there are no redundant documents in the chain; (4) the chain is coherent, that is, two adjacent documents are on the same topic (if they are not on the same topic, they are still related within the context of the input); (5) after reading the chain, new relations among news actors (people, organizations, and places) are learned.

All questions have text answers that are given in positiveness order, which are mapped to an integer scale of 2, 1, -1, and -2. The average of all annotators is taken for each question. The neutral choice of zero is not given to make annotators think more critically, and prevent selecting the first alternative choice that has the minimum cognitive requirements [34]. The last question has two answers, for having the ability to disclose relations or not, mapped to 1 and -1.

2.6.1.4 Annotators and Outlier Elimination

All tasks are assigned to 20 annotators in the first, and 12 in the second user study. Annotators are mostly graduate students, and a few undergraduates and faculty members. In order to estimate the consistency among annotators and detect outliers, we calculate Fleiss kappa [35] for each of the evaluation measures.

For the first user study, we have initially 27 annotators. We calculate Fleiss' kappa with all annotators for each of the evaluation measures. Then, the kappa is recalculated without each annotator. The scores obtained without individual annotators are given to the box-plot method. For each of the evaluation measures, the outlier, or the one with the maximum value if there is no outlier, given by the box-plot is removed from the annotators list. This results in 20 annotators. Shahaf and Guestrin employ 18 annotators; Zhu and Oates do not report the number of annotators. Even then, just for the annotations of the ability to disclose relations question, 6 of 20 annotators are removed due to their misinterpretation of the question that we identify

with a post-survey. After removing outliers according to our heuristics, the kappa scores are increased for all evaluation measures.

2.6.1.5 Consistency Among Annotators

For the first user study, Fleiss’ kappa [35] score is 0.49 for relevance, 0.46 for coverage, 0.12 for redundancy, 0.30 for coherence, and 0.33 for ability to disclose relations. For the second, the same scores are 0.63, 0.34, 0.02, 0.23, and 0.18. Since both redundancy scores are below 0.20, meaning slight agreement among annotators, according to the interpretation of [36], we ignore the results of redundancy.

2.6.2 User Study 1: Varying Framework Parameters

2.6.2.1 Methodology

The first user study consists of 24 chains, obtained by the framework algorithm, with eight sets of parameters (versions A to H) on three topics. The design of this user study, given in Table 2.5, is based on the first four research questions asked in Introduction.

In the decisions, we compare the performance of two or more versions to answer their respective questions. Decisions are independent of each other, i.e. a decision result is not used in later decisions. For a fair evaluation, all parameters, except the one we want to gauge its effect on the algorithm, are kept the same. Based on the observations in preliminary experiments, fixed parameters are selected as all members, vector space model, and a window length of 15 days.

In preliminary experiments, we observe that long chains are overwhelming to

Table 2.5: The design of User Study 1: Eight versions (A to H) of the framework algorithm. *ALL*: the all-members method, *SN*: social network, *VSM*: the vector space model.

(Decision No.) Research Question	Version	Sub-Methods in the Framework		
		Similarity of Candidate with News Chain	Similarity Between News Articles	Window Length in Days
(Decision 1) Is there any proper window length?	A	ALL	VSM	7
	B	ALL	VSM	15
	C	ALL	VSM	30
(Decision 2) Which method for candidate similarity?	B	ALL	VSM	15
	D	Weighted members	VSM	15
(Decision 3) What type of social network?	E	ALL	SN (all actors)	15
	F	ALL	SN (top 500 actors)	15
	B	ALL	VSM	15
(Decision 4) Which method for document similarity?	E	ALL	SN (all actors)	15
	G	ALL	Named entity	15
	H	ALL	Hybrid (SN and VSM)	15

comprehend. In both user studies, we use a heuristic approach that searches for effective similarity-threshold values (θ) in a greedy fashion, by incrementing with a constant value. For the sake of simplicity, it keeps the chain lengths to 15 or fewer documents. The chain length decreases as the threshold values increase. For instance in the hybrid algorithm, we reduce the chain length from 23 to 15 by incrementing the cosine and social-network thresholds from 0.155 to 0.160, and from 0.115 to 0.120, respectively. On average, there are 12 news articles in a chain (median: 13, minimum: 4, maximum: 15).

2.6.2.2 Results of User Study 1

Decision 1: Is there any proper time-window length? While scanning with zigzagged search, we employ a sliding window that captures news articles. In Figure 2.7, we examine three window lengths of 7, 15, and 30 days.

We observe that the performance of varying the window length is case-dependent. For cases with a uniformly distributed number of documents (*Trucks Going to Syria* and *Allegations to Fenerbahçe*), the window should be small—7 days in both cases—in order to not miss news articles in a dense collection. For non-uniformly distributed number of documents (*Ukrainian Riots*), the window should be large (30 days), to catch news articles in a sparse collection.

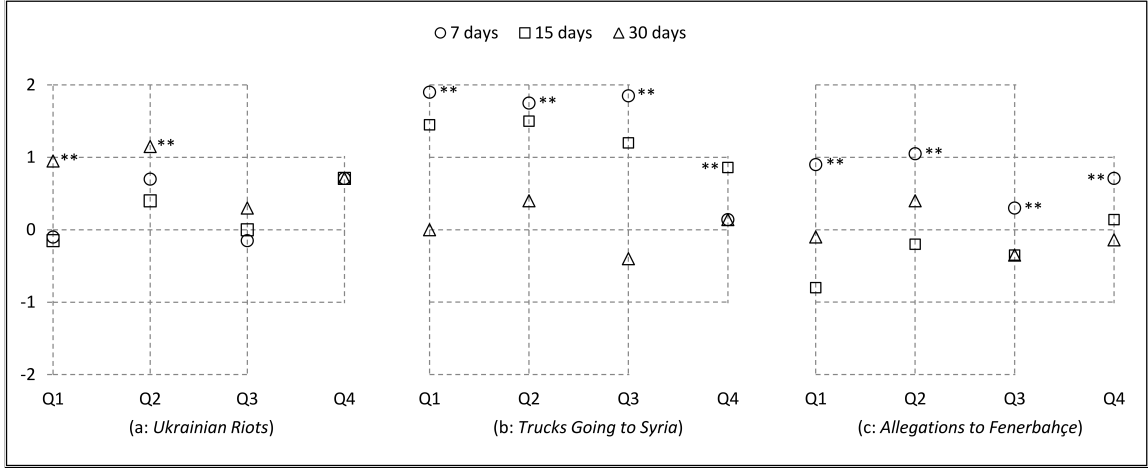


Figure 2.7: Annotation results for Decision 1: Proper window length. Sub-figures (a, b, and c) are for the results of three topics. Question numbers are given in horizontal axis (Q1: *relevance*, Q2: *coverage*, Q3: *coherence*, and Q4: *ability to disclose relations*). The vertical axis represents an average score of annotation answers (scale is between -2 and 2 for Q1-Q3, -1 and 1 for Q4). For the pairwise comparison of the top two algorithms, “**” means that there is a statistically significant increase at the 1% level ($p < .01$), after the corresponding method is applied (see Table 2.6 for details). The same notation is used in the following figures.

In order to test whether a case is uniformly distributed, we apply the Shapiro-Wilks test [37] that states that, with small p -values, the collection does not follow a uniform distribution. In order to apply the test, we divide the collection into intervals of 20 days, and count the number of articles for each case. For *Trucks Going to Syria*, *Allegations to Fenerbahçe*, and *Ukrainian Riots*, p -values are 0.30, 0.50, and 0.10, respectively; *Ukrainian Riots* seems to be less uniformly-distributed than *Trucks Going to Syria* and *Allegations to Fenerbahçe*.

The Friedman test [38] is applied to the results of Figure 2.7; the details are given in Table 2.6. The Friedman test shows if there is a significant difference between at least two methods. This test is applied when there are more than two methods (groups), annotator answers are ordinal-categorical, annotations (observations) are paired and non-uniformly distributed. We use the one-tailed p -values instead of the two tailed, since we try to show that the effectiveness of one algorithm is greater

Table 2.6: The details of the Friedman test for Decision 1 with respect to Figure 2.7. The Friedman tests p -values are listed with Chi-square values. “Pw.d” is the mean difference between the top two algorithms. The p -values of the pairwise comparisons of the top two algorithms are marked in Figure 2.7. The same notation is used in the following tables.

Q: <i>measure</i>	Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe		
	p	Chi.	Pw. d	p	Chi.	Pw. d	p	Chi.	Pw. d
Q1: <i>relevance</i>	<0.01	16.93	1.05	<0.01	29.66	0.55	<0.01	23.35	1.00
Q2: <i>coverage</i>	<0.01	9.46	0.45	<0.01	28.17	0.25	<0.01	19.73	0.65
Q3: <i>coherence</i>	-	-	0.30	<0.01	29.38	0.65	0.012	8.70	0.65
Q4: <i>disclose relations</i>	-	-	-	0.011	8.93	0.72	<0.01	14.71	0.57

than the effectiveness of the others, instead of them being equal. All statistical tests in this study are conducted in the same manner.

In order to have pairwise comparisons, we further apply the post-hoc test proposed by Conover [39], which is valid if the Friedman test indicates any significance. From Figure 2.7 to 2.10, the Conover test results are given for only the top two algorithms, since we want to see the significance of the winner. The scores of the Conover tests are provided in the details web page.

Decision 2: Which method for candidate similarity works better? The effectiveness of the all-members and weighted-members methods depends on the *freshness* of the input, as depicted in Figure 2.8. An input is fresh if it is close to the beginning of the topic. Note that the input documents of all cases are from January 2014. The *Trucks Going to Syria* event starts in January 2014, and *Ukrainian Riots* in November 2013. *Allegations to Fenerbahçe* is relatively old, beginning in July 2011; the weighted-members method works better for this case, since it gives lower importance to old members of the chain including the input that is not fresh. For other cases that we have relatively fresher inputs, the all-members method is more effective in

terms of *relevance*, *coherence*, and *ability to disclose relations*, since it gives the same importance to all members of the chain including the fresh input. Our expectation of weighted members providing more coherent chains fails in some cases.

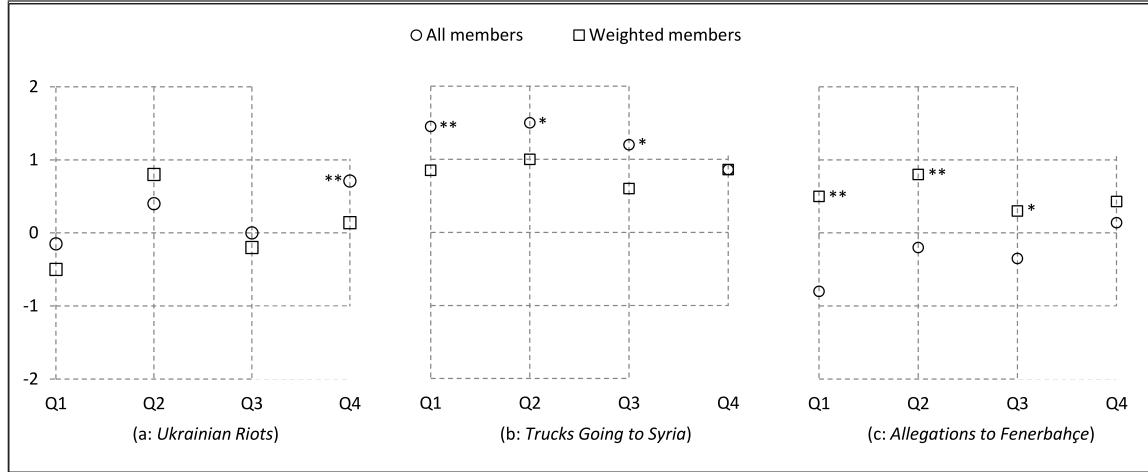


Figure 2.8: Annotation results for Decision 2: *all members* vs. *weighted members*. Note that “*” means that there is a statistically significant increase at the 5% level ($p < .05$), after the corresponding method is applied (see Table 2.7 for details). The same notation is used in the following figures.

The Wilcoxon signed-ranks test [40] is applied to the results of Figure 2.8 to see any significant difference between algorithms; the details are given in Table 2.7. This test is used when there are two methods, annotator answers are ordinal-categorical, and annotations are paired and non-uniformly distributed.

Decision 3: What size of social network works better? The results of using all news actors and the top 500 most important ones, in terms of frequency, are given in Figure 2.9. We observe that using all news actors—approximately 5,000—is more effective for *Trucks Going to Syria*, which has more number of minor actors that are observed with less frequency in the collection. For two cases with more number of major actors (*Allegations to Fenerbahçe* and *Ukrainian Riots*), using the top 500 important news actors is more effective. Using all news actors for such cases reduces

Table 2.7: The details of the Wilcoxon test for Decision 2 with respect to Figure 2.8, where “d” is mean difference, “p”, “Z,” and “r” are scores of the Wilcoxon test. The same notation is used in the following similar tables when the Wilcoxon test is applied.

Q: <i>measure</i>	Ukrainian Riots				Trucks Going to Syria				Allegations to Fenerbahçe			
	d	p	Z	r	d	p	Z	r	d	p	Z	r
Q1: <i>relevance</i>	0.35	0.074	-2.46	-0.93	0.60	0.003	-2.75	-0.92	1.30	< 0.001	-3.72	-1.03
Q2: <i>coverage</i>	0.40	-	-	-	0.50	0.011	-2.29	-0.66	1.00	< 0.001	-3.12	-0.99
Q3: <i>coherence</i>	0.20	-	-	-	0.60	0.033	-1.88	-0.54	0.65	0.020	-2.05	-0.68
Q4: <i>disclose relations</i>	0.57	0.003	-2.75	-0.97	-	-	-	-	0.29	-	-	-

the effectiveness scores, due to possible inclusion of redundant ones. The Wilcoxon test is applied to the results of Figure 2.9; the details are given in Table 2.8.

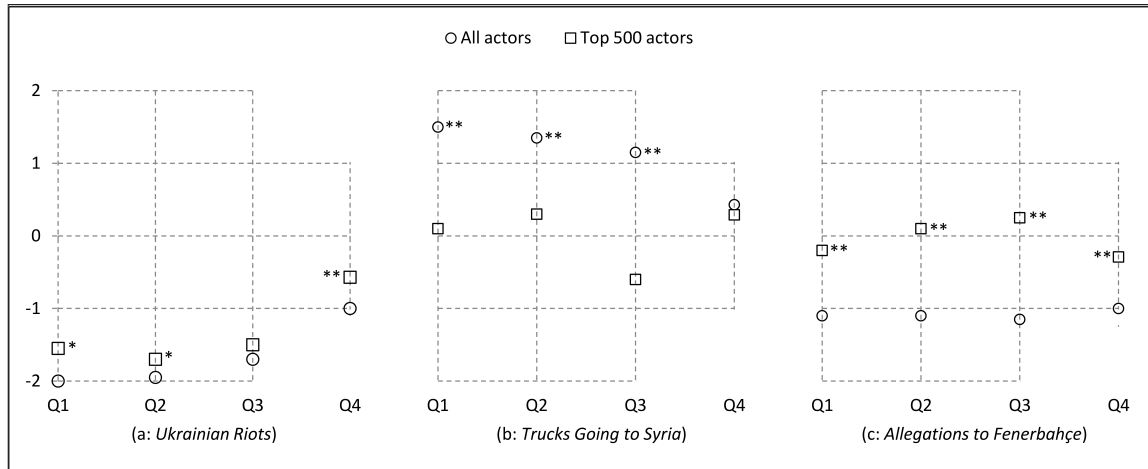


Figure 2.9: Annotation results for Decision 3: *All news actors vs. top 500 news actors*. See Table 2.8 for details of statistical tests.

Decision 4: Which method for document similarity works better? The results of four similarity methods are given in Figure 2.10. The success of the hybrid model, which employs both lexical features and news actors, is case dependent. For the cases

Table 2.8: The details of the Wilcoxon test for Decision 3 with respect to Figure 2.9.

Q: <i>measure</i>	Ukrainian Riots				Trucks Going to Syria				Allegations to Fenerbahçe			
	d	p	Z	r	d	p	Z	r	d	p	Z	r
Q1: <i>relevance</i>	0.45	0.013	-2.23	-0.91	1.40	< 0.001	-3.29	-0.88	0.90	0.002	-2.88	-0.91
Q2: <i>coverage</i>	0.25	0.036	-1.80	-0.68	1.05	0.004	-2.65	-0.73	1.20	< 0.001	-3.29	-0.91
Q3: <i>coherence</i>	0.20	-	-	-	1.75	< 0.001	-3.29	-0.75	1.40	< 0.001	-3.19	-0.85
Q4: <i>disclose relations</i>	0.43	0.009	-2.36	-0.96	0.14	-	-	-	0.71	0.005	-2.58	-0.97

with a relatively higher number of major actors (*Ukrainian Riots* and *Allegations to Fenerbahçe*), the effectiveness of the vector space model is increased by the hybrid model—the only exception is ability to disclose relations of *Allegations to Fenerbahçe*. For the case with a relatively higher number of minor actors (*Trucks Going to Syria*), the effectiveness of the vector space model is not increased by employing news actors.

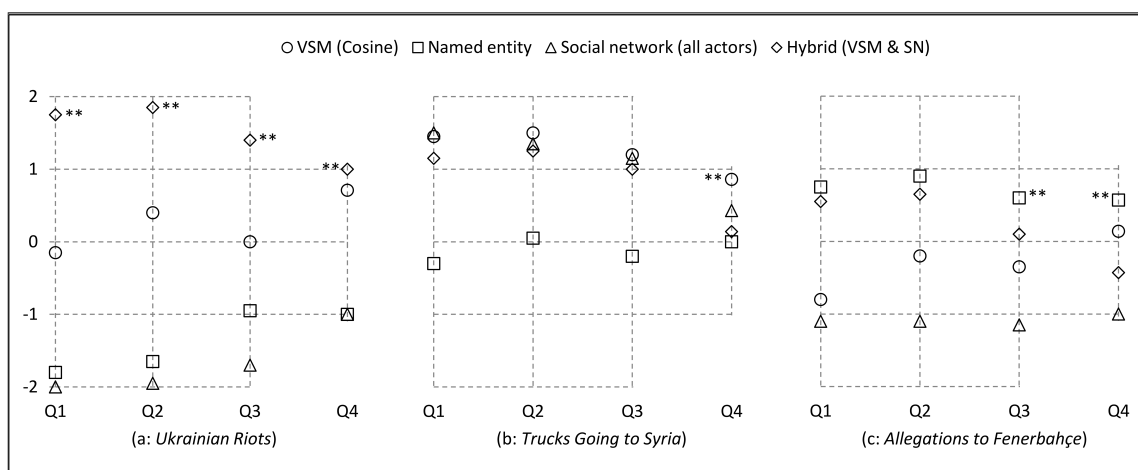


Figure 2.10: Annotation results for Decision 4: Similarity methods. See Table 2.9 for details of statistical tests.

Another observation is that using only named entities—as observed in the TDT

Table 2.9: The details of the Friedman test for Decision 4 with respect to Figure 2.10.

Q: <i>measure</i>	Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe		
	p	Chi.	Pw. d	p	Chi.	Pw. d	p	Chi.	Pw. d
Q1: <i>relevance</i>	<0.01	56.02	1.85	<0.01	35.25	0.05	<0.01	37.05	0.20
Q2: <i>coverage</i>	<0.01	54.73	1.45	<0.01	23.96	0.15	<0.01	35.39	0.25
Q3: <i>coherence</i>	<0.01	41.79	1.40	<0.01	19.20	0.05	<0.01	25.70	0.35
Q4: <i>disclose relations</i>	<0.01	52.42	0.29	<0.01	19.19	0.43	<0.01	25.03	0.43

domain [8, 41]—or only social networks performs poorly. However, the named entity method is more effective than the other methods, in the case of *Allegations to Fenerbahçe*. This can be explained by the fact that this case mostly involves the actor, *Aziz Yıldırım*, who is the club chairman, and not involved in any other case in the given collection. When the case involves many actors, as in *Ukrainian Riots* and *Trucks Going to Syria*, we observe that the effectiveness of using a social network, revealing relations among news actors, is higher than the effectiveness of using only named entities. The Friedman test is applied to the results of Figure 2.10; the details are given in Table 2.9.

2.6.2.3 Recommendations

Based on the results of the first user study, for parameter tuning, we recommend the use of:

1. *Dynamic window length*: When news articles are uniformly distributed, the window should be small. It should be large for non-uniformly distributed cases.
2. *Case-dependent candidate-similarity method*: The weighted-members method

works better for inputs that are not fresh, while the all-members method is more effective with relatively fresher inputs.

3. *Variable social-network size*: For improving efficiency, the size of a social network can be relatively small for cases with a higher number of major actors.
4. *Case-dependent document-similarity method*: Lexical features based on the vector space model are more effective in measuring similarity for cases with minor news actors. When a few number of major actors are involved, the performance of news actor methods can be competitive with the vector space model. The effectiveness of the vector space model and news actors can be improved by combining them in a hybrid model.

2.6.3 User Study 2: Comparison with Baselines

2.6.3.1 Methodology

For comparison, we need to select a representative version of our framework algorithm. We can apply our fine-tuning recommendations on each topic; however, to provide a fair evaluation, we use the same version. Since our contribution is to employ news actors and zigzagged search for story-chain discovery, we choose among versions that employ news actors (named entity, social network, and hybrid). Since using only named entities or only social networks has poor performance, we compare the hybrid version with three baselines—referred to as *hZZ: Hybrid and Zigzagged Search*. The design of User Study 2 is given in Table 2.10.

The first baseline is a simple TDT [7] approach, which examines all documents once, and adds a document to the chain by measuring the cosine similarity with the seed, i.e. input document. The second is an adaptive TDT [42] approach, which is similar to simple TDT, except that it employs a window to scan documents, updating

Table 2.10: The design of User Study 2: Comparing our framework algorithm, *hZZ*: Hybrid and Zigzagged Search, with three baselines, *sTDT*: Simple TDT, *aTDT*: Adaptive TDT, *GN*: Google News.

Method Name	Scanning the Collection	Similarity of Candidate with News Chain	Similarity Between News Articles	Window Length in Days
sTDT	One pass with no window	Only with input document	VSM	-
aTDT	One pass with window	ALL	VSM	15
GN	Unknown			
hZZ	Zigzagged with window	ALL	Hybrid (SN & VSM)	15

the event description after processing each window. This method is similar to our framework, but without using zigzagged search and news actors. In both methods, chain lengths are set to 15 or fewer documents.

The third baseline is the search result list of Google News (<http://news.google.com>). The collection of Google News is a superset of our collection, since it includes *Sözcü* news. The title of the input document is given as a query string. For a fair comparison, we set the range of documents starting from the input date to the end date of our collection, and create a chain with the result list sorted in time. In *Allegations to Fenerbahçe*, since the list includes 40 documents (more than 15), we select (11) equally spaced news articles.

2.6.3.2 Results of User Study 2

The average scores are given in Table 2.11. Scores for each annotator are given in Appendix A. Also, output story chains that are obtained by *hZZ* are listed in Appendix A.

In Table 2.12, the Friedman test is applied to show if there is a significant difference

between at least two methods. Methods are further pairwise compared with the Conover post-hoc test in Table 2.13. In order to measure the effect size of pairwise comparisons, we apply Cohen’s d-test [43]. We highlight cells of Table 2.13 with dark gray if there is a large effect size, and light gray if medium; it remains white if it has a small effect size. The Cohen’s d values and confidence intervals are provided in the details web page.

In total, there are 72 pairwise comparisons between the methods. We have 36 pairwise comparisons in the rows of *hZZ*, which uses zigzagged search and a social network of news actors. The results show that it has statistically significantly higher *relevance* (67% of pairwise comparisons of *hZZ*), *coverage* (56% of pairs), *coherence* (78% of pairs), and *ability to disclose relations* (44% of pairs). We observe that our framework can be helpful to news consumers, since *hZZ* significantly improves effectiveness with respect to baselines, in 61% of pairs (22 of 36 pairs); in the remainder, none of the baselines significantly outperforms our method. All of these pairs have medium (4 of 22 pairs) or large (18 of 22) effect sizes, according to the thresholds of Cohen [44]. Furthermore, we have medium effect sizes in two non-significant additional pairs.

2.7 Discussion

2.7.1 Practical Considerations

We employ the *hZZ* algorithm in the Bilkent News Portal (<http://newsportal.bilkent.edu.tr>), which aggregates Turkish news articles from various resources [29]. We integrate three social-network versions that include different numbers of news actors, by transforming them into matrices of news actors that involve edge weights. Sample screenshots of the system are given in Figure 2.11. The top screen is the front

Table 2.11: The average scores of all annotators for *sTDT*: Simple TDT, *aTDT*: Adaptive TDT, *GN*: Google News, *hZZ*: Hybrid and Zigzagged. The method(s) with the highest score is marked as bold.

Q: <i>measure</i>	Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe					
	sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ
Q1: <i>relevance</i>	1.25	-1.25	0.33	1.67	-0.83	0.83	0.67	1.08	-1.50	-0.83	0.92	1.00
Q2: <i>coverage</i>	1.42	-0.33	0.83	1.67	-0.25	0.83	1.17	1.25	-1.00	-0.83	0.67	0.67
Q3: <i>coherence</i>	0.08	-1.25	0.08	1.33	-1.33	0.33	-0.58	0.42	-1.75	-0.83	-0.08	0.00
Q4: <i>disclose relations</i>	0.83	0.00	0.83	0.83	-0.17	0.50	0.83	0.83	-0.33	0.00	0.83	0.67

Table 2.12: The details of the Friedman test with respect to Table 2.11.

Q: <i>measure</i>	Ukrainian Riots		Trucks Going to Syria		Allegations to Fenerbahçe	
	p	Chi.	p	Chi.	p	Chi.
Q1: <i>relevance</i>	<0.01	25.33	<0.01	20.68	<0.01	28.21
Q2: <i>coverage</i>	<0.01	17.35	<0.01	17.28	<0.01	24.38
Q3: <i>coherence</i>	<0.01	20.76	<0.01	15.72	<0.01	25.07
Q4: <i>disclose relations</i>	<0.05	10.71	<0.01	12.00	<0.01	14.56


page of the portal, where the link of the news-chain discovery tool is provided in the left menu. The bottom screen is where users enter parameters for the algorithm, such as the input document or similarity threshold values.

We observe that mining a large collection can be time-consuming, as experienced in [45], and [15]. To overcome this scaling problem, we ask the user to enter some keywords about the input document, and hence get a subset of news articles to be processed. A similar approach is also applied in the related studies.

Quality of output chains is input-dependent: selecting low similarity thresholds can result in long and noisy chains. Different input documents may require different parameter values.

Table 2.13: Pairwise comparisons of the methods in Table 2.11. Each cell includes the mean difference between the method scores, and the p -value of the Conover test if the difference is statistically significant. Note that “**” and “***” mean that there is a statistically significant increase at the 5% level ($p < .05$), and 1% level ($p < .01$), respectively. Large effect size is indicated with dark gray, and medium with light gray; small remains white.

Q: measure	Case			Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe			
	Method	sTDT	aTDT	GN	sTDT	aTDT	GN	sTDT	aTDT	GN	sTDT	aTDT	GN
Q1: relevance	aTDT	-2.50**			+1.66**			+0.67**					
	GN	-0.92**	+1.58**		+1.50**	-0.16		+2.42**	+1.75**			+1.75**	
	hZZ	+0.42*	+2.92**	+1.34**	+1.91**	+0.25	+0.41	+2.50**	+1.83**			+1.83**	+0.08
Q2: coverage	aTDT	-1.75**			+1.08**			+0.17					
	GN	-0.59	+1.16**		+1.42**	+0.34		+1.67**	+1.50**			+1.50**	
	hZZ	+0.25	+2.00**	+0.84*	+1.50**	+0.42	+0.08	+1.67**	+1.50**			+1.50**	0.00
Q3: coherence	aTDT	-1.33**			+1.66**			+0.92**					
	GN	0.00	+1.33**		+0.75**	-0.91**		+1.67**	+0.75**			+0.75**	
	hZZ	+1.25**	+2.58**	+1.25**	+1.75**	+0.09	+1.00**	+1.75**	+0.83**			+0.83**	+0.08
Q4: disclose relations	aTDT	-0.83**			+0.67**			+0.33					
	GN	0.00	+0.83**		+1.00**	+0.33		+1.16**	+0.83**			+0.83**	
	hZZ	0.00	+0.83**	0.00	+1.00**	+0.33	0.00	+1.00**	+0.67**			+0.67**	-0.16



Bilkent News Portal
BILKENT INFORMATION RETRIEVAL GROUP 13.05.16

[Home](#) | [Help](#) | [About](#)

[TR] [EN] [CREATE ACCOUNT](#) [LOGIN](#)

Turkish News Search

Search News:

CATEGORIES

Economy

Politics

Turkey

World

Sports

Entertainment

Health

Science & Tech

Columns

News Chain Discovery

LATEST NEWS

" ANKARA MERKEZLİ 35 İLDE KPSS"
TRT / KULTURSANAT / 2016-05-13 13:16:43
2010 KPSS soruşturması kapsamında Ankara merkezli 35 ilde eş zamanlı gerçekleştirilen operasyonda, haklarında gözaltı kararı [More...](#)

" ÇUKURCA'DA ÇATIŞMA: 8 SEHİT"
TRT / KULTURSANAT / 2016-05-13 13:16:25...
Hakkari'nin Çukurca ilçesinde teröristlerle çıkan silahlî çatışmada 6 asker, müdahale için bölgeye sevk edilen helikopterin [More...](#)

" SANCAKTEPE SALDIRISIYLA İLGİLİ 8 GÖZALTI"
TRT / KULTURSANAT / 2016-05-13 13:15:29
Sancaktepe'de bombalı araçla gerçekleştirilen terör saldırısına ilişkin İstanbul'da 70 adrese düzenlenen

RECENT & PAST EVENTS

Recent Events	Past Events
TRT TÜRKİYE'NİN YENİ HAVAALANI... TRACKINGS (9)	
" HAVI MARMARA'DA ÖLENLERİN AİLELERİNE... TRACKINGS (8)	
BAKAN YILDIZ: YUMRUK DAVASINDAN... TRACKINGS (8)	
DİZİ SEKTÖRÜ KRİZDE YÜZDE... TRACKINGS (5)	
TERCİHLERDE İL, İLÇE SINIRI... TRACKINGS (22)	
BDP'DEN BAŞBUĞ'A ÇOK SERT... TRACKINGS (11)	
İPTAL AYI'NIN VARLIĞINI TARTIŞMAYA... TRACKINGS (10)	
KİTÇ'DE SENDİKALAR YİNE KUR'AN... TRACKINGS (20)	
AKP'LI BAŞKAN KÜRT SORUNUNA... TRACKINGS (14)	
TOKİ'NİN İSTIKRARLI PROJESİ ATAKENT... TRACKINGS (6)	
ÜÇ BAKANLIKTAN PITBULLI GENELGESİ... TRACKINGS (37)	
PEHLİVANLARIN KAYITLARI BAŞLADI... TRACKINGS (7)	
MUCİDİN SİR ÖLÜMÜ: CİNAYET... TRACKINGS (5)	
NAZARBAYEV'E "ATATÜRK" TEŞEKKÜRÜ... TRACKINGS (23)	
VEKİL İÇLİ: HUKUK DİŞLİK... TRACKINGS (8)	
"HSYK VE YARGITAY'DA İLHAM... TRACKINGS (25)	
BİDEN, İRAN'A İSRAİL GARANTİSİ... TRACKINGS (11)	
"BOMBA YÜKLÜ MINİBÜS'E MÜTALAA... TRACKINGS (6)	
BARROSO'NUN GÖZÜ KULAĞI KIBRIS'TA... TRACKINGS (56)	

Bilkent University - Information Retrieval Group

News-Chain Discovery - Parameter Selection

This page is to select parameters for news-chain discovery, which is supported by TÜBİTAK project no. 113E249.
Please click on "Start News-Chain Discovery" button after entering parameters.

(Note that news-chain discovery will take several minutes if an old news article is selected.)

Select News Date:

Select News ID & Title:

Enter cosine threshold (between 0.0 and 1.0, default value is given):

Enter social network (co-occurrence) threshold:

Figure 2.11: Screenshots (*top*: front page, *down*: user interface for parameter selection) from Bilkent News Portal where our framework for story-chain discovery is applied.

2.7.2 Complexity Analysis

Let N be the number of documents in the collection, w be the number of documents in a window, y be the expected number of documents that are added to chain in zig (forward) phase, and z be the expected number of documents that are added to chain in zag (backward) phase.

In order to find documents to be added to a chain with zigzagged search, all

window members are compared with the current chain. That is, there are w similarity scores calculated between a chain member and the window in the zig (forward) phase, and $w - y$ similarity scores calculated in the zag (backward) phase; total of $2 \times w - y$ comparisons are done between a chain member and window. This is calculated for one chain member. Total number of comparisons for all chain members is the current chain size multiplied by $2 \times w - y$. For the first window, the chain size is 1, therefore $2w - y$ comparisons are done. For the other windows, the chain size is the size of the previous chain plus the expected number of documents that are added to chain in zig (forward) and zag (backward) phases, y and z , respectively.

We assume that the expected numbers of documents that are added to chain in zig and zag phases decrease as window slides. This assumption is based on the following fact. Similarity between chain and candidate document depends on all members of current chain. As window slides, current chain is extended, and the probability of exceeding this similarity becomes smaller. We denote this probability with α . That is, y and z decrease by α as window slides. The number of total comparisons are calculated as follows.

For the first window: $2w - y$

For the second window: $(1 + y + z)(2w - y)$

For the third window: $(1 + (y + \alpha y) + (z + \alpha z))(2w - y)$

For the fourth window: $(1 + (y + \alpha y + \alpha^2 y) + (z + \alpha z + \alpha^2 z))(2w - y)$

...

For the last window: $[1 + (y + \alpha y + \dots + \alpha^{\frac{N}{w}-2} y) + (z + \alpha z + \dots + \alpha^{\frac{N}{w}-2} z)](2w - y)$

The total number of comparisons is $[\frac{N}{w} + [(\frac{N}{w} - 1)y + (\frac{N}{w} - 2)\alpha y + \dots + (\frac{N}{w} - (\frac{N}{w} - 1))\alpha^{\frac{N}{w}-2} y] + [(\frac{N}{w} - 1)z + (\frac{N}{w} - 2)\alpha z + \dots + (\frac{N}{w} - (\frac{N}{w} - 1))\alpha^{\frac{N}{w}-2} z]](2w - y)$

Consequently, we simplify the total number of comparisons as follows.

$$= \left[\frac{N}{w} + (y+z) \left[\sum_{k=0}^{\frac{N}{w}-2} \left(\frac{N}{w} - k - 1 \right) \alpha^k \right] \right] (2w - y)$$

We can further evaluate this equation as follows.

$$= \left[\frac{N}{w} + (2w - y)(y+z)(-1) \left[\left(\sum_{k=0}^{\frac{N}{w}-2} (k+1) \alpha^k \right) - \left(\sum_{k=0}^{\frac{N}{w}-2} \frac{N}{w} \alpha^k \right) \right] \right]$$

The first summation term is an arithmetic-geometric series and the second one is a geometric series. Thus, the equation becomes the following.

$$\begin{aligned} &= \left[\frac{N}{w} + (2w - y)(y+z)(-1) \left[\left(\frac{1 - \frac{N}{w} \alpha^{\left(\frac{N}{w} - 1 \right)} + \left(\frac{N}{w} - 1 \right) \alpha^{\frac{N}{w}}}{(1-\alpha)^2} \right) - \left(\frac{1 - \alpha^{\left(\frac{N}{w} - 1 \right)}}{1-\alpha} \right) \right] \right] \\ &= \left[\frac{N}{w} + (2w - y)(y+z) \left[\frac{N}{w} \alpha^{\left(\frac{N}{w} - 1 \right)} - \left(\frac{N}{w} - 1 \right) \alpha^{\frac{N}{w}} + \frac{1 - \alpha^{\left(\frac{N}{w} - 1 \right)}}{1-\alpha} - 1 \right] \right] \end{aligned}$$

The asymptotical upper bound for this equation is $\mathcal{O}((y+z)N)$. Since $z < y$ and $y+z < 2y$, then $\mathcal{O}(yN)$.

Chapter 3

The Present: Selecting Public Front-page¹

3.1 Motivation

The front page of a news aggregator, like Google News (<http://news.google.com>) or Yahoo! News (<http://news.yahoo.com>), is the showcase where readers expect to see significant news articles. With human-editor-based news aggregators, the burden of reading several news articles and selecting important ones is a challenging task. Editors may select worthless news unintentionally, or even according to their own points of view. As a result, intelligent algorithms that allow news aggregators to process news and select significant ones, accordingly, need to be developed.

¹This study is published in [46].

3.2 Aim

Given a news stream, M , that arrives periodically to a news aggregator, let s be the length of its front page (i.e., number of news articles in the front page); the problem is to select a set of important news articles $I \subseteq M, |I| = s$, which we call front-page news selection.

What are interesting and important news? What makes news interesting or important? These questions are beyond the scope of this study. The Community of Social Sciences tries to answer such questions. Eilders [47] states that readers favor articles with high values of news factors. News factors are some characteristics such as unexpectedness, cultural proximity, and reference to persons, etc. [48]. In this study, we simplify and generalize news factors as follows:

1. *Importance ranking.* News should be ranked according to their importance. Importance is an abstract concept for public front pages. Popularity is one possible measure to quantify importance. Another interpretation of importance is to what degree a news article represents a cluster or topic.
2. *Diversification.* News agenda should be presented with as many viewpoints as possible. Viewpoints can be news categories (classes) or topics.
3. *Length of the front page.* News aggregators have a limited space to present the most important news articles, while diversifying content as much as possible.

There are two types of news selection (news recommendation): personalized and public. Personalized news selection [49] aims at providing news according to the user's interest. A profile model associated with the user is typically generated and candidate news articles are filtered through the profile model whenever the user logs in to the system. The user's past history and similar users' system activities

are exploited to generate the model. Public news aggregators simply assume that popular news articles are important. In this study, we examine news selection for public front pages. Popularity is mostly measured by meta-features, like number of clicks. Selecting important news using click counts is called click-based news selection. However, the number of clicks for a news article is counted during a long period of time, and is therefore not suitable for detecting breaking news. Moreover, the number of clicks cannot be quantified in environments that do not keep track of clicks. Thus, rather than meta-features, we focus on raw text (news content).

News articles in front pages can be diversified by using their category or topic tags. However, our aim is not to use meta-attributes, but to leverage raw text for this purpose. Finding well-separated clusters, or topics, of news articles can be a solution while using only raw text. Topic modelling approaches [50] find separate clusters of documents for different topics and generate topic-word distributions. These words can be used to measure document and topic importance, while choosing different documents from varying topics to provide diversification. We present a novel approach that employs Latent Dirichlet Allocation (LDA) [51] to find diversified public front pages, while taking into consideration the importance of news within topics. LDA is a probabilistic topic-modelling algorithm that finds latent topics in a given text collection, and has been widely applied to various domains such as genetics and computer vision [50].

In the next section, we give related work on news selection, and the diversity of selections. We present our algorithm, and evaluate it based on a user study. We discuss challenges in selecting public front-page news in the same section. Lastly, we conclude this chapter with some future research pointers in public front-page news selection.

3.3 Contributions

The contributions of this study are as follows.

1. A novel algorithm to select public front-page news is introduced that considers the importance, diversity, and length of the front page. We measure document importance and topic importance based on a statistical model that uses topic modelling to provide diversification. We select important documents from important topics using a priority-based method, to fit news content into the length of the front-page.
2. To the best of our knowledge, this is the first study that examines public front-page news selection using only raw text.
3. We conduct a user study and measure the effectiveness and diversity of our algorithm, with our new annotation program. Annotation results show that up to 7 of 10 news articles are tagged as important and up to 9 of them are from different topics.

3.4 Related Work

3.4.1 News Selection

To the best of our knowledge, news selection for public front pages, using only raw text, has not been studied before. However, there is a study that compares public front pages chosen by news editors and the interest of the social media crowd; but it does not present an algorithm for our task [52].

The task is similar to the traditional information-retrieval task, in which a set of documents and a query are given, and a subset of documents related to the query are returned by ranking according to their relevance to the query. The difference is that there is no query for selecting front-page news.

Selecting a subset from a document collection is a general task; we assume news recommendation is the most related research area to front-page news selection. Recommendation systems are mainly divided into three categories [53]:

In content-based recommendation, news content that is clicked or favored by users is processed, and then news content that is similar to favored ones is recommended. In its simplest form, similarity among news content is measured with metrics such as the cosine similarity [54]. There is also the content-based filtering approach, which creates a user profile implicitly or explicitly, and filters other news content according to the user profile. Getting feedback from news readers is one way of explicit user profiling [55]. Other studies track user activities to create a user profile implicitly [56].

Collaborative filtering aims to exploit similar users' activity on the system. A typical example for news recommendation using collaborative filtering is the early version of Google News [57]. Users with similar click history are fetched and news articles they read are recommended. Later, the recommendation approach of Google News changed to adapt both collaborative and content-based filtering [49]. The content-based approach that models users' information profiles is mixed with the previous collaborative, click-history method. The user profile is built on her news interests by using news articles that she read before. This is an example of hybrid recommendation that aims to combine both advantages of content-based and collaborative method to provide more effective systems [58].

In terms of target community, recommendation systems are divided into two categories: personal and public recommendation. The former considers only a specific user while deciding on a recommendation. Collaborative filtering is a method of personal recommendation. The latter is harder to solve than personal recommendation, since there are many different user needs waiting to be satisfied. Content-based approaches can yield a solution for public recommendation.

The algorithm introduced in this study is an instance of content-based public recommendation. It does not spy on user activities, nor does it get feedback from users. Instead, the raw text of news articles are processed without any user information or meta-features, like click counts.

3.4.2 Diversity in Document Selection

There are algorithms for selecting a diversified set of documents among a given collection, based on measures such as maximal marginal relevance [59]. The application of such algorithms is examined in [60]. Selection based on diversity is also studied in the context of publish/subscribe systems [61]. In such systems, documents are obtained in a given period of time and then a subset of them is selected by applying greedy search, to find the most diverse item. Recommendation systems that consider diversification [62] find documents of a similar interest to the user profile/query, while presenting diverse results from various topics.

In this study, we do not apply such approaches directly, since our algorithm handles different factors altogether; namely, importance, diversity, and the length of the front page. For diversification, we utilize topic modelling that aims at finding latent groups/topics in a text collection. To the best of our knowledge, there is no study adapting topic modelling for public front-page news selection.

3.5 Front-page News Selection based on Topic Modelling

An overview of our front-page news selection approach is given in Figure 3.1. Main steps of our approach can be summarized as follows (each step is detailed in the following sub-sections):

1. Given a news stream with M documents, find topics, with topic modelling, to provide diversity, and assign each document to a topic.
2. Find the importance of documents and rank documents in each topic according to document importance.
3. Find the importance of topics.
4. Select the most important news articles, in the most important topics, with a priority-based method for fitting to the length of the front page.

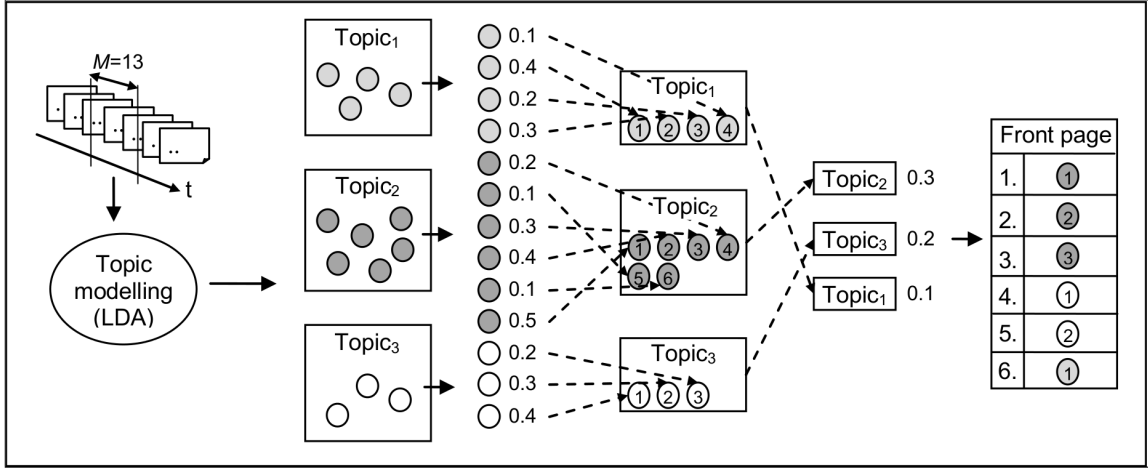


Figure 3.1: Overview of our front-page news selection approach. A news stream with $M = 13$ documents and a front page with a length of 6 are given as an example. Documents are represented with circles and topics with rectangles. The numbers on the right of circles and rectangles are imaginary importance values for documents and topics, respectively. From left to right, dashed lines are used for document and topic ranking.

3.5.1 Finding Topics

The LDA algorithm assigns latent topics to each word in a given text collection [51]. Briefly, LDA outputs topic-word (ϕ) and document-topic (θ) distributions. The word distribution for topic c (ϕ_c) estimates the probability of a word being generated by topic c . The topic distribution for document d (θ_d) estimates the probability of a topic being generated by document d . These are used for determining the topic of a document, and also the representative words for each topic. LDA learns a topic model, including these distributions, and can be used to predict the topic of a new document. However, in this study, we aim at utilizing distributions obtained from the model to estimate the importance of documents and topics.

3.5.2 Finding Document Importance

Let the number of documents in a given text collection be M , d_i 's topic be c , and assume that the importance value for the document d_i ($1 \leq i \leq M$) is estimated by using ϕ_c of each word included in d_i . This measures, intuitively, d_i 's importance by calculating how words in d_i represent d_i 's topic. Let the topic assigned to d_i (i.e. highest probability in θ_i) be c , the number of words in d_i be T_i , and t_{ij} be a word in d_i ($1 \leq j \leq T_i$); then the weight of word t_{ij} in ϕ_c is w_{ij} . The following function—*doc_imp(.)* measures how important (representative) a document is for its topic:

$$doc_imp(d_i) = \frac{\sum_{j=1}^{T_i} w_{ij}}{T_i} \quad (1 \leq i \leq M) \quad (3.1)$$

We observe that a small number of words have high weights while others have low weights, which implies the power law [63]. It suggests we trim low-weighted words, which are unimportant in the context of a given document, while calculating the *doc_imp(.)* function. We verify if ϕ follows a power-law distribution with the goodness-of-fit test [33]. The hypothesis of this test claims ϕ is generated from a power-law distribution, and the test outputs a p -value that can be used for quantifying the validity of the hypothesis. If p -value is smaller than 0.1, then the hypothesis is rejected, which means ϕ does not fit into a power-law distribution. If it is higher than 0.1, then ϕ is plausible for fitting a power-law distribution. For an arbitrarily chosen topic obtained from a random subset of a news collection that is used in evaluation, the p -value is obtained as 0.42. We observe similar patterns in other topics as well, and thus conclude that p plausibly fits into a power law.

For trimming unimportant words, the above equation is modified to quantify only for words in d_i that are seen in top- k words of ϕ_c as follows:

$$\begin{aligned}
doc_imp(d_i) &= \frac{\sum_{j=1}^{T_i} w_{ij} \times \alpha}{T_i} \\
\alpha &= \begin{cases} 1, & \text{if } t_{ij} \text{ is in top-}k \text{ of } \phi_c \\ 0, & \text{otherwise} \end{cases} \quad (1 \leq i \leq M)
\end{aligned} \tag{3.2}$$

Note that the denominator is still document size, instead of the number of words seen in top- k of ϕ_c , which is to avoid overvaluing long documents with repeated high-weighted words (remember that T_i is the length of d_i). We determine the value of k as 20% of distribution length, which is obtained by the Pareto principle, which implies the 80-20 law [63]. This means that 80% of words that have a high weight are in the first 20% of ϕ_c .

3.5.3 Finding Topic Importance

Let S be the number of topics, which is given as an input to LDA. There are studies to determine the number of topics in a text collection [64]; but for simplicity, assuming the number of documents in the given collection is M , we simply assume S 's value is adapted from clustering studies [65] as $S = \sqrt{M/2}$ (rounded to the nearest integer). Can and Ozkarahan [66] propose $m \times n / t$, where m is the number of documents, n is the number of terms, and t is the number of non-zero cells in the document-term matrix. We do not adapt this approach, since we observe in our preliminary experiments that both methods have similar results, and the calculation of the former method is simpler than of the latter.

For each topic c_i ($1 \leq i \leq S$), topic importance is calculated with the weights of words in ϕ_{c_i} and the importance values of documents that are assigned to c_i . Note that the same words appear in ϕ of all topics; but the weights of words in each ϕ

are not necessarily the same. Let R be the total number of unique words in the given collection, r_{ij} be a unique word in topic c_i ($1 \leq j \leq R$), D_i be the total number of documents in topic c_i , d_{ij} be a document in topic c_i ($1 \leq j \leq D_i$), then the weight of r_{ij} in ϕ_c is w_{ij} . Assume k is obtained by the Pareto principle as explained in the previous subsection. The importance of topic c_i is calculated with the *topic_imp(.)* function as follows:

$$\begin{aligned}
 \text{topic_imp}(c_i) &= \frac{\sum_{j=1}^R w_{ij} \times \alpha}{R} + \frac{\sum_{j=1}^{D_i} \text{doc_imp}(d_{ij})}{D_i} \\
 \alpha &= \begin{cases} 1, & \text{if } r_{ij} \text{ is in top-}k \text{ of } \phi_{c_i} \\ 0, & \text{otherwise} \end{cases} \quad (1 \leq i \leq T)
 \end{aligned} \tag{3.3}$$

The first summation term in the above equation estimates the importance of top- k words of c_i by summing up their weights in ϕ_{c_i} . The second summation term estimates the importance of documents in c_i by summing up their *doc_imp(.)* values. Note that the number of words selected from R is exactly k , while the number of words selected from T_i in *doc_imp(.)* is equal to or lower than k , since a document may not necessarily include all top- k words of its topic-word distribution.

Figure 3.2 shows an example calculation to find document and topic importance by Equations 3.2 and 3.3, respectively.

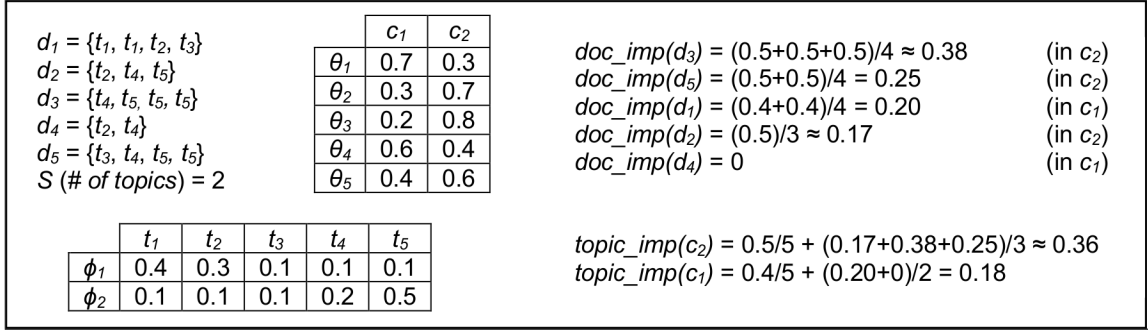


Figure 3.2: A sample document collection with 5 documents, 2 topics, and 5 unique words is given to demonstrate how to find document and topic importance. Recall that we use 20% of words in ϕ_c , then $k = 1$ for 5 unique words.

3.5.4 Priority-based News Selection using Document and Topic Importance

Having estimated topic and document importance values, we apply our priority-based method to select important news articles from various topics for a front page with a certain length.

In operating systems, priority scheduling [67] solves the problem that the CPU must serve waiting processes in a limited time. Each process has a priority and time length. The CPU starts with the process with the highest priority and serves until it finishes. Other processes are then served in the same manner. In this study, we simulate that the CPU is our algorithm for public front-page news selection, and a process is a topic. Each topic has a demand of placing its most important news articles on front-page. Our approach decides to serve important news articles in a topic by considering the topic's priority, demand, and length of the front page.

Each topic has a priority value for being selected for the front page. Priority of c_i is calculated with the function $topic_pri(.)$ as the portion of its importance value

over all topic importance values:

$$topic_pri(c_i) = \frac{topic_imp(c_i)}{\sum_{j=1}^S topic_imp(c_j)} \quad (1 \leq i \leq S) \quad (3.4)$$

Each topic also has a demand that shows how many important news articles this topic would like to place onto the front page. The demand of c_i is calculated with the function $topic_dem(\cdot)$ —rounded to the nearest integer—where h is the constant to represent a news article’s share in the front page and is calculated as $h = 1/f$ where f is the length of the front page:

$$topic_dem(c_i) = \frac{topic_pri(c_i)}{h} \quad (1 \leq i \leq S) \quad (3.5)$$

For the top place(s), our approach selects the most important news article(s) of the topic, which has(have) the highest priority, ordered by document importance. Other slots are served by topic priorities and their demands.

Assume Figure 3.3 shows our front-page news selection strategy based on the same news collection given in Figure 3.2 and front-page length (f) is set to 3. At the top of Figure 3.3, $topic_pri(\cdot)$ and $topic_dem(\cdot)$ values are calculated. For instance, $topic_pri(c_1) = 0.33$ means that the first topic has a share of 33% on the front page and $topic_dem(c_1) = 1$ means that the first topic demands 1 news article, according to its weight of importance. Since the highest topic importance value belongs to the second topic (inferred from the weight of importance values), we serve all of its demands in the first two slots of the front page. Then, demand of the first topic is served in the remaining one slot. Note that each topic has a document ranking, according to document importance by Equation 3.2, and d_3 and d_5 are the most important two news articles in the second topic while d_1 is the most important one in the first topic.

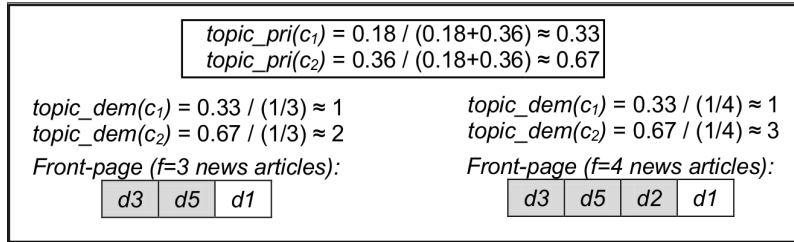


Figure 3.3: An illustration of selecting news for two different front-page lengths, based on the same news collection given in Figure 3.2. Here, $\text{topic_pri}(\cdot)$ estimates priority of a topic and $\text{topic_dem}(\cdot)$ finds how many documents a topic demands to place on front page. In the front-page representation, documents of the second topic are shaded with gray.

Our front-page news selection approach is given in Figure 3.4.

```

Input: News collection with  $M$  documents, front-page length  $f$ .
Output: Front-page  $F[\dots]$  that includes  $f$  news articles in ranked order.

// Find topics (see Section 3.5.1)
1  Number of topics,  $S \leftarrow \sqrt{M/2}$ 
2  Get topic-word distribution ( $\phi$ ) and document-topic distribution ( $\theta$ ) by topic modelling
// Find importance of documents (see Section 3.5.2)
3  for  $i=1 \dots M$  do
4      Assign document  $d_i$  to the topic with the highest weight in  $\theta_{d_i}$ 
5      Get importance of  $d_i$  // (see Eq. 3.2)
6  end-for
// Find importance of topics (see Section 3.5.3)
7  for  $i=1 \dots S$  do
8      Get importance of topic  $c_i$  // (see Eq. 3.3)
9      Rank documents in  $c_i$  according to their importance
10 end-for
// Find priority and demand of topics (see Section 3.5.4)
11 for  $i=1 \dots S$  do
12     Get priority and demand of  $c_i$  // (see Eq. 3.4 and 3.5)
13 end-for
// Select  $f$  news articles for front-page (see Section 3.5.4)
// Consider topics in a ranked order according to their priority
14 while  $F$  has empty slot do
15     Consider the next topic
16      $k \leftarrow$  demand of this topic // (see Eq. 3.5)
17     Place top- $k$  news articles from this topic to  $F$ 
// Note that available empty slots in  $F$  can be less than  $k$ 
// For example, if there are two slots available and  $k=3$ , then select the top-2
18 end-while

```

Figure 3.4: Pseudocode for our front-page news selection approach.

3.6 Evaluation

To the best of our knowledge, there is no gold-standard dataset of news articles, with labels of importance; therefore, we conduct a user study to evaluate the effectiveness of our algorithm in terms of document importance and topic diversity. Since we are not aware of any news selection baseline algorithm for public front pages, with which we can compare our algorithm, the user study evaluates only our front-page news selection algorithm. However, we have an additional user study to find the effectiveness of random news selection, and compare it with our algorithm. In this section, we explain the details and results of our user study that aims to evaluate our front-page news selection algorithm by using a newly generated annotation program.

3.6.1 Setup

Approaches used to select front-page news are evaluated over a dataset that includes labels of whether a document is important or not; so, traditional metrics such as precision and recall can be measured. However, we are not aware of any labelled dataset suitable for our task. Yahoo! published a test collection including number of clicks for news articles in Yahoo! Front-page Today Module [68]; however this dataset does not include news article content and thus, cannot be used in our study and similar studies that would examine methods using news content. Instead we use a non-labelled news collection, including 15,844 news articles that were obtained from the Milliyet (<http://www.milliyet.com.tr>) newspaper on 36 different days between 09/09/2009 and 31/10/2009, in which we obtain the cleanest raw text, via the Bilkent News Portal (<http://newsportal.bilkent.edu.tr>). Since news articles were obtained from a real-time RSS resource, the number of news articles for each day differs.

News articles are first pre-processed by stemming and removing stopwords that

are common words in Turkish. The stemming strategy is to use just the first five characters of all words, which is shown to yield good results in Turkish text [30]. The stopword list used in this study is a slightly extended version of the one obtained from [29]. The LDA algorithm is implemented with MALLET library [69] into our approach.

Since there are 36 days in the news collection, we run our algorithm for each day and get 36 different front pages. The front-page length is set to 10. We then conduct a user study to capture the importance and diversity of news articles selected by our algorithm.

3.6.2 User Study

We create an annotation program, with a user-friendly interface, capable of providing the label importance, while maintaining the diversity of the front page. A total of 19 graduate and undergraduate students are selected to be volunteer annotators. Before starting the annotation process, all annotators read the user manual that explains background information about the domain, tasks they will encounter, and how to use the annotation program.

Figure 3.5 shows a sample screenshot from the annotation program. In the top-left panel, news articles are listed by their snippets of 200 characters. Full text is accessible upon double-clicking on snippets. Annotators have two tasks that must be accomplished consecutively:

1. First, annotators have a binary decision; whether given news articles are important or not, based on their own interests in the right panel.
2. After deciding for all news articles in the right panel, they determine which

news article belongs to which topic, i.e. the diversity of a front page, using the bottom-left panel. However, there are no pre-defined topics in this task. Instead, they drag and drop each news article into a virtual cluster whose aim is to collect news articles belonging to the same topic. We call them virtual clusters since they have no descriptive title. Annotators add new, delete existing, or change contents of virtual clusters if needed.

In the annotation screen of Figure 3.5, the importance annotation is completed since all news articles are labelled as important/unimportant in the right panel; however, diversity annotation is not finished. Four news articles are put into three virtual clusters (shaded with dark gray in the right panel), and six news articles are still waiting for drag-dropping.

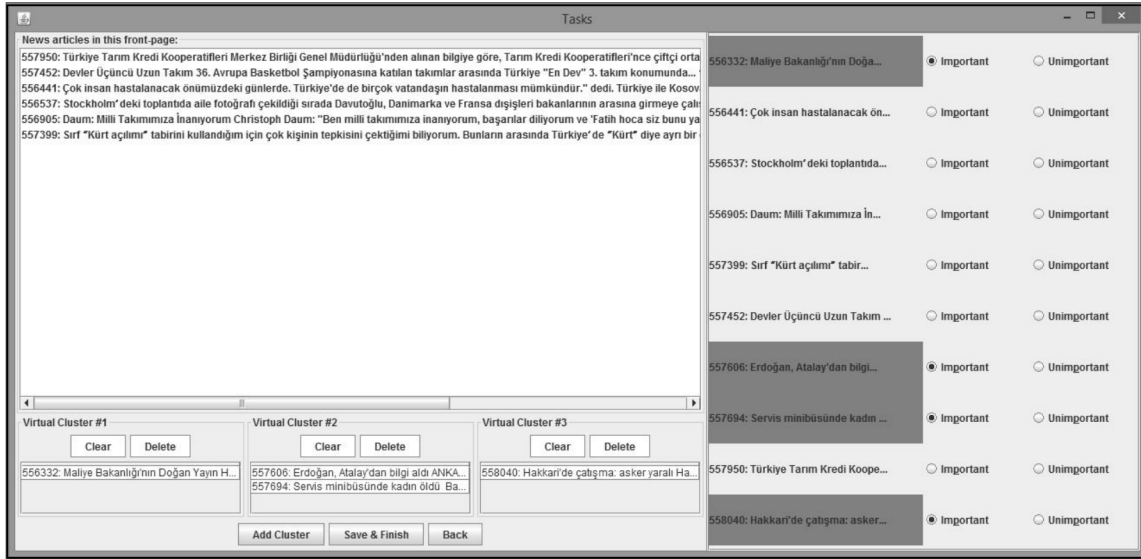


Figure 3.5: A sample screenshot from the annotation program. The front page has 10 news articles, and initially all are listed in the top-left panel. Annotators are asked to (1) assess the importance of each news article in the right panel and then (2) drag and drop each news article into a virtual cluster (topic) in the bottom-left panel. In this figure, all news articles are labelled as important or unimportant. However, diversity annotation is not yet completed. Four news articles (shaded with gray in the right panel) are dropped into three virtual clusters, and six news articles are still waiting for drag-dropping.

In the user study, each annotator is assigned to the same tasks. Annotators have to complete 36 different front-page annotation processes. They may quit the program during a task and continue from the same state of annotation whenever they would like to do so. Also they can redo annotations that they finished earlier.

After all annotations are done, the importance and diversity of a front page are measured by $ann_imp(.)$ and $ann_div(.)$, respectively. Let p_d be the total number of positive decisions (i.e. important news articles), where d is a day in the given news collection. For the annotator a , importance of a front page of d , with length s , is $imp_a(d) = p_d/s$. Then, the importance of all front pages annotated by a is $ann_imp(a) = \sum_{i=1}^{36} imp_a(d_i)/36$.

Similarly, let c_d be the total number of virtual clusters for day d . For the annotator a , diversity of a front page of d , with length s , is $div_a(d) = c_d/s$. Then, the diversity of all front pages annotated by a is $ann_div(a) = \sum_{i=1}^{36} div_a(d_i)/36$. Both $ann_imp(.)$ and $ann_div(.)$ have a value between 0 and 1. The higher the importance and diversity annotators achieve, the more important and diverse front pages our algorithm is meant to find.

Since there are more than two annotators, we calculate Fleiss Kappa [35] to estimate consistency between annotators. For importance and diversity annotations, Fleiss' Kappa is calculated 0.29 and 0.09, respectively. According to the interpretation given by [36], there is a fair agreement for annotations of importance while annotations of diversity have slight agreement. Not having a strong agreement can be attributed to differences among user interests, which is an expected observation in public front-page news selection.

Table 3.1: Average, median, standard deviation, minimum, and maximum of *ann-imp* and *ann-div* scores are listed for the user study of 19 annotators.

Type	Avg.	Median	Std.Dev.	Min.	Max.
<i>ann-imp</i> (annotator importance)	0.52	0.51	0.10	0.27	0.70
<i>ann-div</i> (annotator diversity)	0.76	0.83	0.15	0.51	0.94

3.6.3 Results and Discussion

Results of the user study are summarized in Table 3.1 that lists the average, median, standard deviation, minimum, and maximum of annotator importance and diversity when 19 annotators are considered. Details of annotation results are given in Appendix B.

In the best case, we provide front pages including up to 70% important news articles, while up to 94% of them belong to different news topics. On the average case, our approach finds front pages including 52% important news articles while 76% of them belong to different news topics.

Note that front-page length is 10 in the user study. Front-pages with a large number of news articles are difficult to annotate for importance and diversity; this is because annotations require reading the content of all news articles. The more news articles that are involved, the more information that should be remembered.

Our results are not compared with any of those of other approaches to select public front-page news, since we are not aware of any similar study or method for our task. Click-based news selection is a possible solution; but no benchmarking is possible, since to the best of our knowledge, there is no dataset that includes both click counts and raw text. However, we can compare the results of our approach with random news selection. For this purpose, in an additional user study, four annotators other than the previous ones are asked to assess the importance of all 15,844 news

articles and only 1,315 (8%) of them are assessed as important. Details of additional annotation results are available in Appendix B. Thus, if random news selection is used, it is expected that approximately 8% of news articles in a front page would be important. It shows that our 52% success rate is 6.5 times more effective than that of random news selection.

One may also think of other solutions for this task, such as applying machine learning. In machine learning, a training model is learned by a classification algorithm, and then documents that have no class information are labelled by the learned model. Such an approach has obstacles. Firstly, if machine learning is used for selecting public front-page news, a test collection, including gold standard is initially needed. However, there is no such test collection for such a task. Moreover, since the number of important news articles are much less than that of unimportant ones, there would be a bias towards unimportant news. Lastly, the news agenda would change in a news portal as new documents arrive. Since previously learned models show decreased performance on recent news agenda, a novel model should be learned based on previous ones. However, only the initial training model is learned with a gold standard. Performance would gradually decrease for recently learned models, based on previous news.

Another possible solution can be click-based news selection. Using this approach, one can measure the popularity of news articles easily by quantifying the number of clicks. However, initially the number of clicks does not exist, and also, misleading click information might be generated by robots, click spam, etc.

Chapter 4

The Future: Filtering Microblogs for Predicting Public Reactions to News¹

4.1 Motivation

Traditional news sources like newspapers provide limited information, due to the drawbacks of slow editorship and time restrictions for reaching event sources. Tweets are recently used as a dynamic news source; a typical example is getting correct updates from disasters [71]. Another kind of information that tweets expose is peoples opinions. With the growth of social media usage, opinions of crowds can be processed to understand mass behaviors, like riots in the Arab Spring [72].

¹A part of this study is published in [70].

4.2 Aim

We define public reaction as peoples acts or behaviors that result from common opinions for an event occurred at a particular time and place. In today’s society of multiple views, it is difficult to estimate the dimension and direction of public reactions for events. For a news article titled “Fire Department saves cat from tree”, no public reaction is expected. On the other hand, mass discontent of people in social media for a news article titled “bloody balance sheet of holiday accidents” is an example for negative reaction. The reaction associated with the accidents does not involve any protests in the streets; however, workers can protest a radical change in an employment law in the streets. In addition to the dimension, its direction (negative vs. positive) is also important. An example of positive reaction is peaceful post-match celebrations of football fans after a championship.

Recent research utilize social media for prediction of consumer behaviors [73], or real-time event detection [74]. Early prediction of public reaction for an event supports government institutions, commercial organizations, and individuals to prepare themselves, in terms of precautions for future negative events, and advantageous responses for future positive events.

In this chapter, we filter a microblog collection, specifically tweets, according to their relevance to news events in order to exploit these tweets for predicting public reactions to the same events. Given a news article as input, we develop a pipeline that starts with fetching at most 5 days of tweets after the origin date of news event. We preprocess tweets, which includes cleaning, normalization, and stemming steps. Tweets are filtered by searching important keywords. Filtered tweets are analyzed in terms of frequency, sentiment, temporal, and spatial features.

In the next section, we list our contributions. We then give a summary of related work, explain our system in details, describe our collection, and conclude the study.

4.3 Contributions

Our contributions are the following. We

1. filter microblog texts, and analyze them according to several features to be used in predicting public reactions,
2. and create a public-reaction dataset that have important events, such as terrorist attacks, in Turkey from 2015 to 2017.

4.4 Related Work

Since our study relies on prediction of future events, we review this section in terms of (1) event detection, (2) prediction with social media, and (3) prediction of public reaction or crowd behavior.

4.4.1 Event Detection

Identifying real-time events using a text stream like tweets is called event detection. Becker et al. [75] cluster tweets to identify events on-the-fly, and then, tweets can be classified as event-related or not. Similar studies try to detect events from tweet streams by employing topic modeling [76] and graph structures [77]. Such studies mostly focus on global and large-scale events like riots. On the other hand, Twitcident [78] detects early signals of real-time incidents that are published by Twitter users, to help crisis management. Jasmine [79] is another system that focuses on local-event detection by exploiting spatial information. [74] detects already-occurred

events of public reactions like riots. [80] detects events by using co-occurrence statistics of words in Turkish tweets. All of these studies are not event predictors, but detectors for already-occurred events that take advantage of dynamic environment of social media.

4.4.2 Prediction with Social Media

In the last decade, prediction with social media is a hot research topic. Researchers mostly try to predict future consumer behaviors like box-office earnings [72], book sales [81], stock market indicators [82], by exploiting tweet features. Some studies do not use features that are extracted from social media, but other sources. For instance, [83] exploits news features to predict news popularity in social media. In this study, we do not predict news popularity, but the dimension and direction of the public reaction to a news event. Given a candidate and target event, [84] predicts the probability of candidate event to cause the target event, by learning correlation between news events reported by the NY Times. They do not focus on early prediction of public reactions. Given an input news article, our system filters tweets and analyze features to be exploited in predicting public-reaction class as target.

4.4.3 Prediction of Public Reactions

Kallus [85] utilizes big data of web including social media to predict crowd behavior such as significant protests. Their prediction is correct when there is a significant protest in the given country during the following three days. Muthiad et al. [86] develops a protest-prediction system that uses several web sources, such as social media and RSS feeds. They apply key-phrase filtering, and extract spatial and temporal information from source documents to generate warnings in advance. Korkmaz et

al. [87] employs an anonymity network, *Tor*, and currency ex-change rates as predictor features to train a logistic regression model. Their predictions are tested for at most following 30 days. Our difference is that we define target public-reaction classes that identify the dimension and direction of a future event, and provide an early-prediction system for a given arbitrary news article.

4.5 Our Filtering System

Our system starts with fetching at most 5 days of tweets after the origin date of news event. We then preprocess tweets, which includes cleaning, normalization, and stemming steps. Filtered tweets are analyzed in terms of frequency, sentiment, temporal, and spatial features.

4.5.1 Preprocessing Tweets

Given a tweet collection, we have the following preprocess operations. Turkish tweets are detected by the language attribute of Twitter API. We divide tweet content into tokens by space character. Noise in tokens, such as characters not in the dictionary, are removed. Numbers in tokens are ignored. Hashtag character is not removed to keep trend topics in inverted index. Invalid emoticon expressions are removed. We create a list of positive, neutral, and negative emoticon characters; and keep them to be exploited in the sentiment-analysis phase.

Turkish has special accent characters, which are 'ç', 'ğ', 'ı', 'ö', 'ş', 'ü'. In microblogs, Turkish users sometimes type the ASCII version of these accent characters, which results in missing or ambiguous words for our sentiment analyzer and indexer.

Replacing ASCII characters with intended original ones is called *deasciification*. Previous studies depends on IV (In Vocabulary) lexicons [88], or training sets [89] with a greedy decision algorithm [90] for deasciification. In this study, we develop a deasciification approach that recursively produces all accent versions of a token, and chooses the one that has the maximum document frequency on the given collection. The idea is the more a token version is used by microblog users, the more potential it is correct. Current approaches mostly use in-vocabulary (IV) lexicon; we do not use it for a couple of reasons. First, we make our algorithm generic to all collections in different languages. Second, we capture the cases that training sets cannot capture, such as a new phrase or slang word is produced by users. We create a lexicon for common abbreviations, and normalize tweets accordingly. We also remove tokens that have length more than 20 characters.

Stems of tokens are obtained by Zemberek [91], which is a popular Turkish stemmer, and *F5* that simply considers first five letters, and performed well in our previous studies [8, 30]. To avoid ambiguity, we consider only the first root that is found by Zemberek. We also apply the stopword list given in the study of [29].

4.5.2 Sentiment Analyzer

In this study, tweet polarity has three directions as positive, negative, and neutral. We find the polarity score of a token with two approaches: *SentiTurkNet* [92] and *SentiStrength* [93].

SentiTurkNet is a Turkish polarity resource that have scores for the synsets in Turkish WordNet. We calculate the sentiment score of a tweet by having the sum of individual polarity scores of its tokens. Since there are no emoticon characters and slang words in *SentiTurkNet*, which are commonly observed in tweets, we manually create a lexicon for them.

SentiStrength is a popular sentiment analyser for English, which can be adapted to other languages by modifying its lexicon files. For emotions and booster words, we use the Turkish lexicon files that are constructed by [94]. Turkish has two negation forms; having negating (1) suffix, e.g. “sevmedi” whose translation is “did not like”, or (2) words after verbs and nouns, e.g. “iyi değil” whose translation is “not good”. We examine words morphologically with Zemberek [91], and add heuristic rules to detect negations. We also detect emoticons by checking our emoticon lexicon, and replace positive ones with “:)”, and negative ones with “:(”. We then give the processed content to *SentiStrength*.

The difference between these two methods is that, in SentiTurkNet, we calculate the sentiment score of a tweet by summing up the polarity of each word in tweet; so, multiple occurrences of a word can boost the score. However, *SentiStrength* always gives a score between -4 and 4. We normalize the output scores of each method to the scale of -1 to 1. Similarly, we observe that *SentiStrength* with modified lexicon files produces more consistent results in our preliminary experiments. At the end, we get a sentiment table, by *SentiStrength*, that includes polarity scores for all tweets to be used in the training phase later.

4.5.3 Filtering

Finding related tweets to a given news article is a challenging task due to a couple of reasons. First, we have a very huge collection of tweets, so with a lazy approach, it would take days to get a subset. Second, directing unrelated tweets to the training phase, i.e false positives, misleads the training model. Considering such concerns, we search tweets by giving an important keyword regarding the news event to the system.

4.6 Analysis

4.6.1 Dataset

We create a new public-reaction dataset, called BilPredict-2017 [95] that consists of three components. First component is the ground truth that has 80 news events, represented by articles, occurred between 2015 and 2017. Each event is listed with its origin date, place, news url, public-reaction category, and reaction tags.

Public reactions are labeled by experts, in terms of dimensions and directions. Labels are suggested by one expert, and controlled and verified by the second expert. Dimensions are in terms of national, local, and social media. National categories represent public reactions occurred in at least two different cities. Local categories have events occurred at only a specific place. Social categories represent reactions that people share opinions only in social media, such as microblogs. Directions are either negative or positive. We have 7 social-reaction categories:

1. *National Negative Reaction*: People react to an event in negative manner over all nation, for instance protests in several cities.
2. *National Positive Reaction*: People react to an event in positive manner over all nation, for instance championship celebrations in several cities.
3. *Local Negative Reaction*: People react to an event in negative manner in a specific place, for instance protest in front of a building.
4. *Local Positive Reaction*: People react to an event in positive manner in a specific place, for instance championship celebration of a local team.
5. *Social-Network Negative Reaction*: People react to an event in negative manner just in Internet, but not in streets.

6. *Social-Network Positive Reaction*: People react to an event in positive manner just in Internet, but not in streets.
7. *No Reaction*.

A sample instance from BilPredict-2017 is given in Appendix C.

The second component is the html file contents of 80 news events. They can be used as input to our prediction system. The last component is the tweets for 80 events. For the next 10 days after the origin date of each news event in BilPredict-2017, we collect approximately 1.3 billions tweets. These tweets can be exploited to create features for prediction models.

4.6.2 Setup

We select one news event from each category in BilPredict-2017, total of 6 events except the category of *No Reaction*. Since our aim is to filter tweets and analyze features for prediction, we just select one representative from each category to analyze filtering and important features. The id numbers and titles of these events in the collection are given in Table 4.1.

Table 4.1: The selected news events for the analysis in filtering.

ID	Title
3	The champions of the 2015 Turkish Super League is Galatasaray.
14	The terror attack in Dağlıca.
17	Aziz Sancar won the Nobel Prize in Chemistry 2015.
43	Alanyaspor qualified to the Turkish Super League.
46	The 10 th Year Anthem is forbidden in Bolu.
61	Magazine programmer confuses Madonna.

Given a news article as input, we analyze 5 days of tweets after the origin date of news event, by dividing them into chunks of 50,000 tweets. These chunks have time lengths of about 15 minutes.

4.6.3 Results and Discussion

Filtered tweets are analyzed in terms of 5 features. These features are (1) the tweet date, (2) frequency of tweets, (3) total sentiment score normalized by frequency, (4) most-frequently observed location in chunks, represented by the order number that they are observed in text, and (5) total follower count normalized by taking its logarithm. Figures 4.1 to 4.6 show the scatter plots of these features for each event, respectively the order in Table 4.1.

In Figure 4.1, we observe that there are ~ 400 chunks of tweets, all chunks have tweets collected for consecutive ~ 15 minutes. The number of tweets are high, at most ~ 350 tweets, at the beginning of the event. Frequency of tweets becomes decreasing as time increases. This relationship is validated with the Spearman's Rank Correlation Coefficient [96]. Spearman's $\rho = -0.40$, where $p < 0.01$. Sentiment of tweets are mostly in the positive scale, since people have celebrations of the championship. There are also some negative tweets on this event. These negative tweets are mostly written by the supporters of rival football clubs. Location of tweet owners is spread over many cities in Turkey, since Galatasaray is a popular football club in Turkey. However, most of tweets are written in İstanbul, which draws a long line at the location number 56. Note that Galatasaray is a football club of İstanbul. For the follower count of tweet owners, we do not observe a specific pattern.

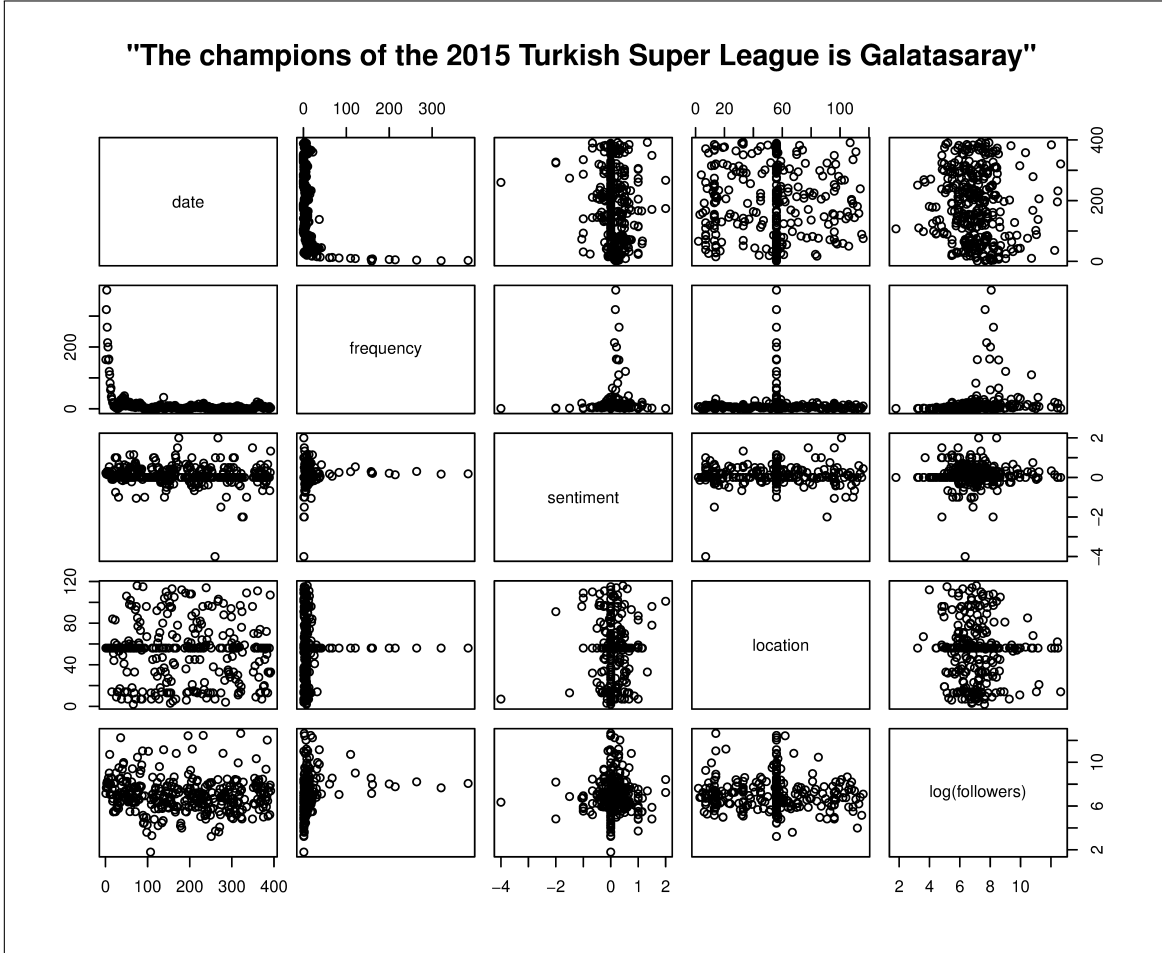


Figure 4.1: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “The champions of the 2015 Turkish Super League is Galatasaray.” The scale of date starts from the beginning of event, ends after almost 5 days. There are ~ 400 chunks of tweets, all chunks have tweets collected for consecutive ~ 15 minutes. The total sentiment scores are normalized by frequency, which results in $[-4.0, 4.0]$ —the scale of *SentiStrength*. The most-frequently observed location in chunks are represented by the order number that they are observed in text. The total number of followers in chunks are normalized by their logarithm. The same notation is used in the following similar figures.

In Figure 4.2, we observe that there are ~ 370 chunks of tweets, coming for consecutive ~ 15 minutes. Frequency of tweets becomes decreasing as time increases. This relationship is validated with the Spearman’s Correlation, where $\rho = -0.72$, where

$p < 0.01$. Sentiment of tweets are mostly in the negative scale, since the event is a terrorist attack. Location of tweet owners is spread over many cities in Turkey, since the terrorist attack is one of the major terror incidents in Turkish history. However, most of tweets are written in İstanbul, probably due to its crowdedness, which draws a long line at the location number 42. For the follower count of tweet owners, interestingly, we observe that the people with a high number of followers have tweets after ~ 24 hours of the beginning of event.

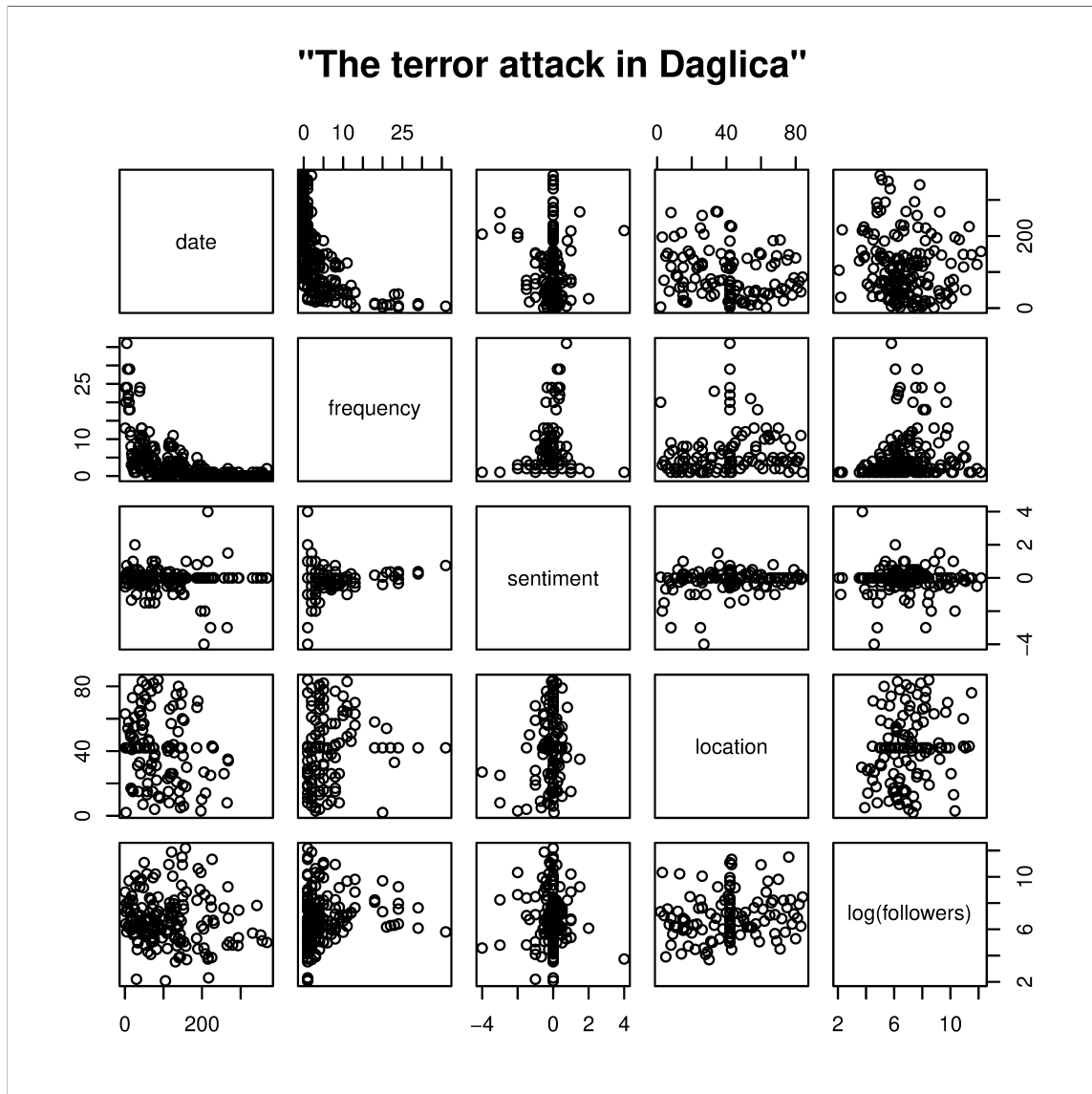


Figure 4.2: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “the terror attack in Dağlica.” The location 42 is İstanbul.

In Figure 4.3, we observe that there are ~ 280 chunks of tweets, coming for consecutive ~ 15 minutes. Since the total number of tweets is less than that of the previous events, the scatter plot is sparse. Frequency of tweets becomes decreasing

as time increases. This relationship is validated with the Spearman's Correlation, where $\rho = -0.65$, where $p < 0.01$. Sentiment of tweets are mostly in the positive scale, since Aziz Sancar is the second Turk in history, who receives the Nobel prize. Location of tweet owners is spread over many cities in Turkey. However, İstanbul has a significant number of tweets, probably due to its crowdedness, which draws a line at the location number 15. For the follower count of tweet owners, we do not observe a specific pattern.

In Figure 4.4, we observe that there are ~ 210 chunks of tweets, coming for consecutive ~ 15 minutes. Since the total number of tweets is less than that of the previous events, the scatter plot is sparse. Frequency of tweets becomes decreasing as time increases. This relationship is not strong, but still validated with the Spearman's Correlation, where $\rho = -0.27$, where $p < 0.01$. Sentiment of tweets are mostly in the positive scale, since this is the first time that Alanyaspor qualifies for the Turkish Super League. Location of tweet owners is not spread over many cities in Turkey, as expected, since Alanyaspor is a football club of Alanya, where is a local city in Turkey. In this case, the top 3 locations are Alanya, Antalya, and İstanbul, which draw a line at the location number 2, 3, and 4, respectively. For the follower count of tweet owners, we do not observe a specific pattern.

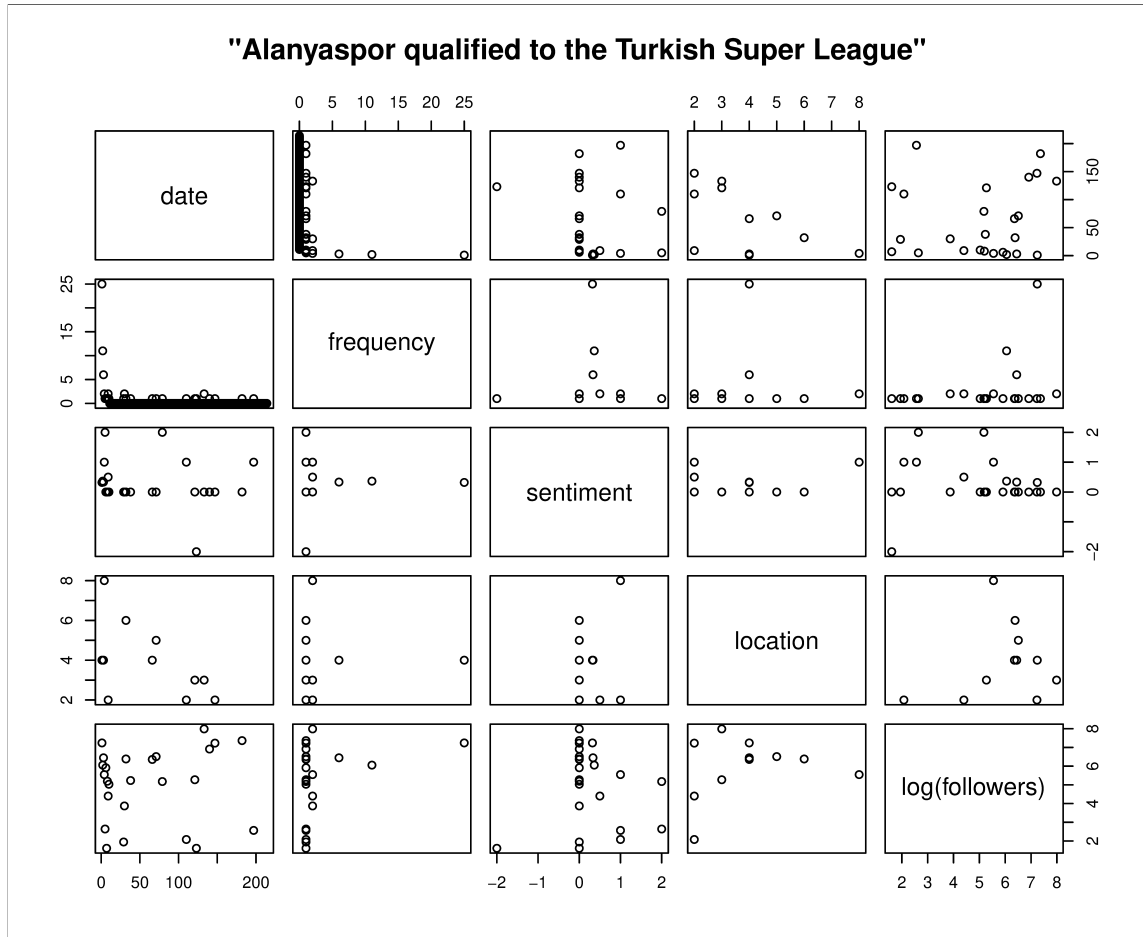


Figure 4.4: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “Alanyaspor qualified to the Turkish Super League.” The location 2, 3, and 4 are Alanya, Antalya, and İstanbul, respectively.

In Figure 4.5, we observe that there are ~ 310 chunks of tweets, coming for consecutive ~ 15 minutes. Frequency of tweets becomes decreasing as time increases. This relationship is not strong, but still validated with the Spearman’s Correlation, where $\rho = -0.30$, where $p < 0.01$. Interestingly, the number of tweets increase again after ~ 3 days of the beginning of events. This is probably due to the fact that some protests occur during those days. Sentiments are mostly in the negative scale during the first days of the event. But after ~ 3 days, there seems to be a division in

sentiments. This is probably due to the conflict of political viewpoints in Turkey. Location of tweet owners is spread over many cities in Turkey, as expected, since the event is popular over all country. For the follower count of tweet owners, we do not observe a specific pattern.

In Figure 4.5, we observe that there are ~ 330 chunks of tweets, coming for consecutive ~ 15 minutes. Frequency of tweets becomes decreasing as time increases. This relationship is validated with the Spearman's Correlation, where $\rho = -0.63$, where $p < 0.01$. Sentiments are mostly in the negative scale, as expected, since the title of a well-known book in Turkey has Madonna, and the magazine programmer thinks wrongly that the book is about Madonna the singer. Location of tweet owners is spread over many cities in Turkey. However, İstanbul has a significant number of tweets, probably due to its crowdedness, which draws a line at the location number 13. We also observe that the people who have high number of followers are attracted to this event in social media.

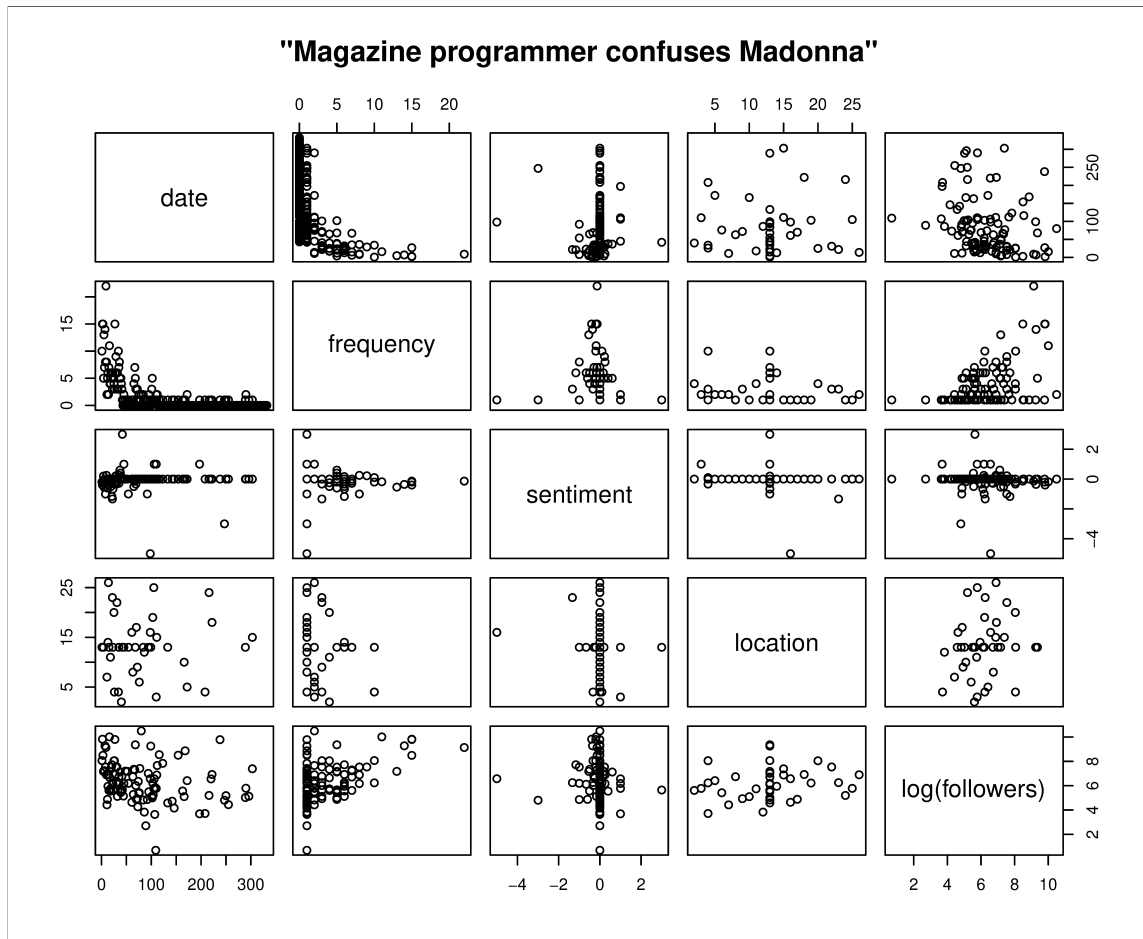


Figure 4.6: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “Magazine programmer confuses Madonna.” The location 13 is İstanbul.

Results show that there is a high correlation between time and frequency of tweets. Sentiment and spatial features also reflect the nature of events, thus all of these features can be utilized in predicting the future.

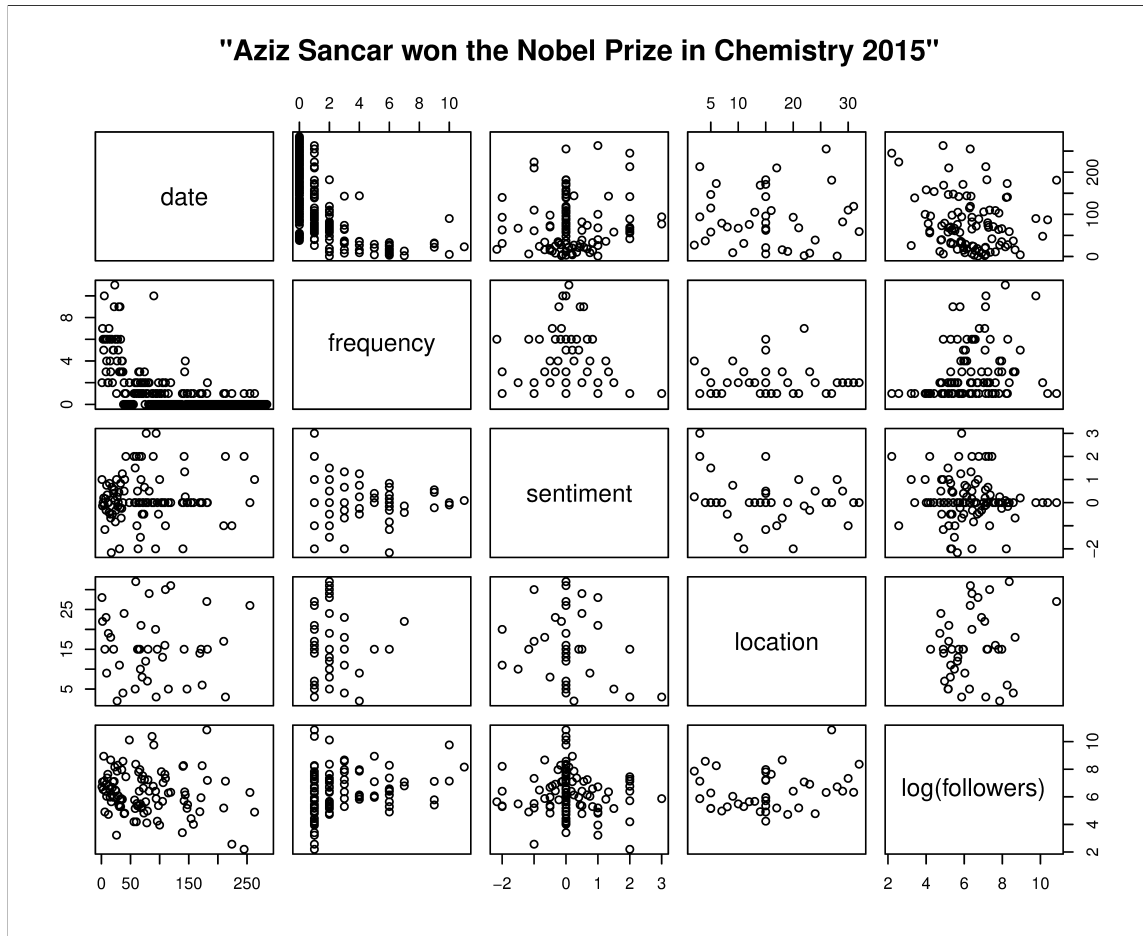


Figure 4.3: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled "Aziz Sancar won the Nobel Prize in Chemistry 2015." The location 15 is İstanbul.

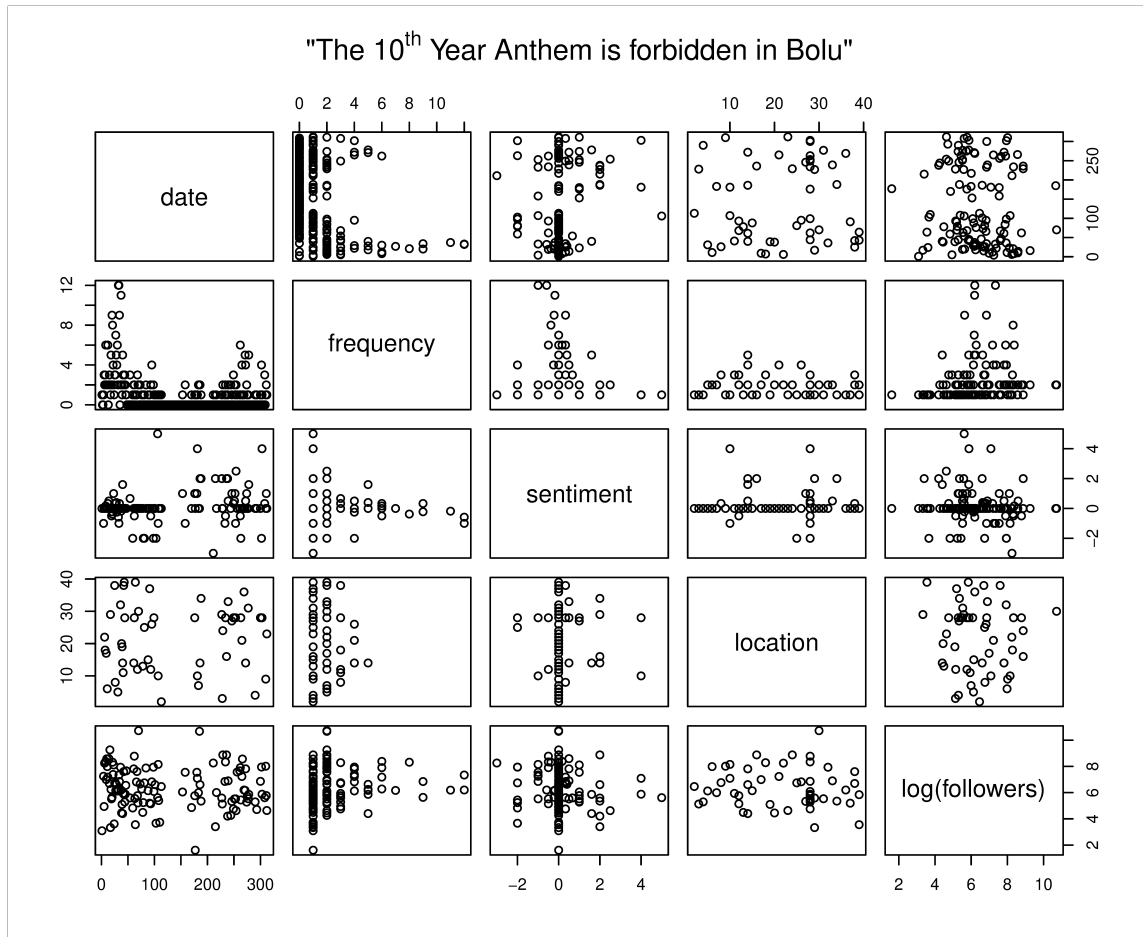


Figure 4.5: The scatter plot of the date, frequency, sentiment, location, and follower of filtered tweets regarding the event titled “the 10th Year Anthem is forbidden in Bolu.” The location 28 and 4 are İstanbul and Ankara, respectively.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis has three parts regarding the past, present, and future of news streams. We discover news chains using zigzagged search in the past, select front-page of current news for the public, and filter tweets with a comprehensive analysis of features to be exploited in predicting future public reactions to events.

In Chapter 2, we present a framework to discover story chains in a given text collection for an input document. We apply a novel text-mining method that uses zigzagged search that reinvestigates past documents based on the updated chain. News actors are utilized to reveal connections among news articles. We conduct two user studies that evaluate our framework in terms of effectiveness. The first compares several versions of the framework to set a guideline for use. The second compares our method with three baselines. The results show that our method provides statistically significant improvement in effectiveness, in 61% of pairwise comparisons,

with medium or large effect size.

In Chapter 3, we develop a novel approach for public news selection by using only raw text. Our method selects the most important news articles in the most important topics with a priority-based method for fitting to the length of the front page. We develop an annotation program for the purpose of conducting a user study. The results show that our topic modelling-based approach for public front-page news selection encourages the use of only raw text. In the best case, 70% of news articles are important, and 94% are of different topics. Moreover, on the average case, 52% of news articles are important; it is about 6.5 times more effective than the 8% success rate of random-news selection. Also, 76% of news articles are of different topics, on the average case.

In Chapter 4, we filter a microblog collection, specifically tweets, according to their relevance to news events in order to exploit these tweets for predicting public reactions to these events. Given a news article as input, we filter tweets by using important keywords. We also create a new collection, called BilPredict-2017, that includes several news events and tweets between 2015 and 2017. Results show that there is a high correlation between time and frequency of tweets. Sentiment and spatial features also reflect the nature of events, thus all of these features can be utilized in predicting the future.

5.2 Future Work

In future work, our framework for discovering story chains can be extended and adopted into other domains that use temporal data, such as the analysis of intelligence reports and micro-blogs. There is also a need for visualization tools that can help users examine chains, and test collections that can help researchers assess and

compare their results.

For selecting public front-pages, topic tracking and novelty detection can be adapted for improving diversity and likewise, named-entity recognition can be used for improving news selection. There is also a need for test collections that include both news content and number of clicks. After providing a proper stopword list and stemmer, our method is language- and domain-independent and is suitable to similar, text-based applications such as blog, review, and intelligence-report aggregators.

We plan to exploit the features that we analyze in filtering microblogs, to predict public reactions to news events. We also plan to extend BilPredict-2017 to include a ground truth that has pairs of tweet and its related news event. The success of filtering and prediction methods can then be evaluated on BilPredict-2017.

Bibliography

- [1] C. Toraman and F. Can, “Discovering story chains: A framework based on zigzagged search and news actors,” *Journal of the Association for Information Science and Technology*, DOI:10.1002/asi.23885, 2017.
- [2] M. S. Hossain, P. Butler, A. P. Boedihardjo, and N. Ramakrishnan, “Storytelling in entity networks to support intelligence analysts,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, (New York, NY, USA), pp. 1375–1383, ACM, 2012.
- [3] Y.-H. Tseng, C.-J. Lin, and Y.-I. Lin, “Text mining techniques for patent analysis,” *Inf. Process. Manage.*, vol. 43, pp. 1216–1247, Sept. 2007.
- [4] A. Stranieri and J. Zeleznikow, *Knowledge Discovery from Legal Databases*. Springer Publishing Company, Incorporated, 1st ed., 2011.
- [5] D. Shahaf and C. Guestrin, “Connecting two (or less) dots: Discovering structure in news articles,” *ACM Transactions on Knowledge Discovery from Data*, vol. 5, pp. 24:1–24:31, Feb. 2012.
- [6] X. Zhu and T. Oates, “Finding story chains in newswire articles using random walks,” *Information Systems Frontiers*, vol. 16, pp. 753–769, Nov. 2014.
- [7] J. Allan, ed., *Topic Detection and Tracking: Event-based Information Organization*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

- [8] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar, “New event detection and topic tracking in Turkish,” *Journal of the Association for Information Science and Technology*, vol. 61, pp. 802–819, Apr. 2010.
- [9] Q. Mei and C. Zhai, “Discovering evolutionary theme patterns from text: An exploration of temporal text mining,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05*, (New York, NY, USA), pp. 198–207, ACM, 2005.
- [10] I. Subašić and B. Berendt, “From bursty patterns to bursty facts: The effectiveness of temporal text mining for news,” in *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, (Amsterdam, The Netherlands, The Netherlands), pp. 517–522, IOS Press, 2010.
- [11] R. Nallapati, A. Feng, F. Peng, and J. Allan, “Event threading within news topics,” in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM '04*, (New York, NY, USA), pp. 446–453, ACM, 2004.
- [12] C. C. Yang, X. Shi, and C.-P. Wei, “Discovering event evolution graphs from news corpora,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 39, pp. 850–863, July 2009.
- [13] D. Kim and A. Oh, “Topic chains for understanding a news corpus,” in *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II, CICLing'11*, (Berlin, Heidelberg), pp. 163–176, Springer-Verlag, 2011.
- [14] J. Song, Y. Huang, X. Qi, Y. Li, F. Li, K. Fu, and T. Huang, “Discovering hierarchical topic evolution in time-stamped documents,” *J. Assoc. Inf. Sci. Technol.*, vol. 67, pp. 915–927, Apr. 2016.

- [15] D. Shahaf, C. Guestrin, E. Horvitz, and J. Leskovec, “Information cartography,” *Communications of the ACM*, vol. 58, pp. 62–73, Oct. 2015.
- [16] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang, “Evolutionary timeline summarization: A balanced optimization framework via iterative substitution,” in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’11, (New York, NY, USA), pp. 745–754, ACM, 2011.
- [17] J. Kleinberg, “Bursty and hierarchical structure in streams,” *Data Mining and Knowledge Discovery*, vol. 7, pp. 373–397, Oct. 2003.
- [18] D. Kumar, N. Ramakrishnan, R. F. Helm, and M. Potts, “Algorithms for storytelling,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 736–751, June 2008.
- [19] R. Choudhary, S. Mehta, A. Bagchi, and R. Balakrishnan, “Towards characterization of actor evolution and interactions in news corpora,” in *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR’08, (Berlin, Heidelberg), pp. 422–429, Springer-Verlag, 2008.
- [20] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*. 2011.
- [21] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL ’05, (Stroudsburg, PA, USA), pp. 363–370, Association for Computational Linguistics, 2005.

- [22] L. Ratinov and D. Roth, “Design challenges and misconceptions in named entity recognition,” in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL ’09, (Stroudsburg, PA, USA), pp. 147–155, Association for Computational Linguistics, 2009.
- [23] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Linguisticae Investigationes*, vol. 30, pp. 3–26, January 2007. Publisher: John Benjamins Publishing Company.
- [24] R. Steinberger, B. Pouliquen, M. Kabadjov, and E. Van der Goot, “Jrc-names: A freely available, highly multilingual named entity resource,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 104–110, 2011.
- [25] G. Tür, D. Hakkani-tür, and K. Oflazer, “A statistical information extraction system for Turkish,” *Natural Language Engineering*, vol. 9, pp. 181–210, June 2003.
- [26] S. Tatar and I. Cicekli, “Automatic rule learning exploiting morphological features for named entity recognition in Turkish,” *J. Inf. Sci.*, vol. 37, pp. 137–151, Apr. 2011.
- [27] G. A. Seker and G. Eryigit, “Initial explorations on using crfs for Turkish named entity recognition,” in *COLING*, pp. 2459–2474, 2012.
- [28] D. Küçük and A. Yazıcı, “Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos,” *Knowledge-Based Systems*, vol. 24, pp. 844–857, Aug. 2011.
- [29] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, and O. M. Vuravas, “Information retrieval on Turkish texts,” *Journal of the Association for Information Science and Technology*, vol. 59, pp. 407–421, Feb. 2008.

- [30] C. Toraman, F. Can, and S. Koberber, “Developing a text categorization template for Turkish news portals,” in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 379–383, June 2011.
- [31] S. Cucerzan, “Large-scale named entity disambiguation based on wikipedia data,” in *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pp. 708–716, 2007.
- [32] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [33] A. Clauset, C. R. Shalizi, and M. E. J. Newman, “Power-law distributions in empirical data,” *SIAM Review*, vol. 51, pp. 661–703, Nov. 2009.
- [34] J. A. Krosnick, A. L. Holbrook, M. K. Berent, R. T. Carson, W. Michael Hanemann, R. J. Kopp, R. Cameron Mitchell, S. Presser, P. A. Ruud, V. Kerry Smith, *et al.*, “The impact of ”no opinion” response options on data quality: Non-attitude reduction or an invitation to satisfice?,” *Public Opinion Quarterly*, vol. 66, no. 3, pp. 371–403, 2002.
- [35] J. L. Fleiss, “Measuring nominal scale agreement among many raters.,” *Psychological bulletin*, vol. 76, no. 5, pp. 378–382, 1971.
- [36] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, vol. 33(1), pp. 159–174, 1977.
- [37] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, pp. 591–611, Dec. 1965.
- [38] M. Friedman, “The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance,” *Journal of the American Statistical Association*, vol. 32, pp. 675–701, Dec. 1937.

- [39] W. Conover, *Practical nonparametric statistics*. Wiley series in probability and statistics, New York, NY [u.a.]: Wiley, 3. ed ed., 1999.
- [40] F. Wilcoxon, “Individual Comparisons by Ranking Methods,” *Biometrics Bulletin*, vol. 1, pp. 80–83, Dec. 1945.
- [41] G. Kumaran and J. Allan, “Text classification and named entities for new event detection,” in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’04, (New York, NY, USA), pp. 297–304, ACM, 2004.
- [42] J. Allan, R. Papka, and V. Lavrenko, “On-line new event detection and tracking,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’98, (New York, NY, USA), pp. 37–45, ACM, 1998.
- [43] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.
- [44] J. Cohen, “A Power Primer,” *Psychological Bulletin*, vol. 112, no. 1, pp. 155–159, 1992.
- [45] D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn, “Newsinessence: Summarizing online news topics,” *Communications of the ACM*, vol. 48, pp. 95–98, Oct. 2005.
- [46] C. Toraman and F. Can, “A front-page news-selection algorithm based on topic modelling using raw text,” *Journal of Information Science*, vol. 41, no. 5, pp. 676–685, 2015.
- [47] C. Eilders, “The role of news factors in media use,” Discussion Papers, Research Unit: The Public and the Social Movement FS III 96-104, Social Science Research Center Berlin (WZB), 1996.

- [48] J. Galtung and M. H. Ruge, “The structure of foreign news: The presentation of the congo, cuba and cyprus crises in four norwegian newspapers,” *Journal of Peace Research*, vol. 2, no. 1, pp. 64–90, 1965.
- [49] J. Liu, P. Dolan, and E. R. Pedersen, “Personalized news recommendation based on click behavior,” in *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI ’10, (New York, NY, USA), pp. 31–40, ACM, 2010.
- [50] D. M. Blei, “Probabilistic topic models,” *Communications of the ACM*, vol. 55, pp. 77–84, Apr. 2012.
- [51] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [52] A. Zubiaga, “Newspaper editors vs the crowd: On the appropriateness of front page news selection,” in *Proceedings of the 22nd International Conference on World Wide Web*, WWW ’13 Companion, (New York, NY, USA), pp. 879–880, ACM, 2013.
- [53] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, pp. 5–53, Jan. 2004.
- [54] J.-w. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn, “Open user profiles for adaptive news systems: Help or harm?,” in *Proceedings of the 16th International Conference on World Wide Web*, WWW ’07, (New York, NY, USA), pp. 11–20, ACM, 2007.
- [55] D. Billsus and M. J. Pazzani, “A hybrid user model for news story classification,” in *Proceedings of the Seventh International Conference on User Modeling*, UM ’99, (Secaucus, NJ, USA), pp. 99–108, Springer-Verlag New York, Inc., 1999.

- [56] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, “Combining collaborative filtering with personal agents for better recommendations,” in *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI ’99/IAAI ’99, (Menlo Park, CA, USA), pp. 439–446, American Association for Artificial Intelligence, 1999.
- [57] A. S. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: Scalable online collaborative filtering,” in *Proceedings of the 16th International Conference on World Wide Web*, WWW ’07, (New York, NY, USA), pp. 271–280, ACM, 2007.
- [58] W. Chu and S.-T. Park, “Personalized recommendation on dynamic content using predictive bilinear models,” in *Proceedings of the 18th International Conference on World Wide Web*, WWW ’09, (New York, NY, USA), pp. 691–700, ACM, 2009.
- [59] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’98, (New York, NY, USA), pp. 335–336, ACM, 1998.
- [60] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, “Novelty and diversity in information retrieval evaluation,” in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’08, (New York, NY, USA), pp. 659–666, ACM, 2008.

- [61] M. Drosou and E. Pitoura, “Dynamic diversification of continuous data,” in *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT ’12, (New York, NY, USA), pp. 216–227, ACM, 2012.
- [62] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in *Proceedings of the 14th International Conference on World Wide Web*, WWW ’05, (New York, NY, USA), pp. 22–32, ACM, 2005.
- [63] M. Newman, “Power laws, pareto distributions and zipf’s law,” *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 2005.
- [64] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical dirichlet processes,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.
- [65] K. Mardia, J. Kent, and J. Bibby, *Multivariate analysis*. Probability and mathematical statistics, London [u.a.]: Acad. Press, 1979.
- [66] F. Can and E. A. Ozkarahan, “Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases,” *ACM Transactions on Database Systems*, vol. 15, pp. 483–517, Dec. 1990.
- [67] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, p.992. Wiley Publishing, 8th ed., 2008.
- [68] Y. Research, “Yahoo! labs datasets front-page today module click dataset.” <http://webscope.sandbox.yahoo.com/catalog.php?datatype=r>, 2015. Accessed: 2017-09-13.
- [69] A. K. McCallum, “Mallet: A machine learning for language toolkit.” <http://mallet.cs.umass.edu/>, 2002. Accessed: 2017-09-13.

- [70] C. Toraman, “Early prediction of public reactions to news events using microblogs,” in *Proceedings of the 7th Symposium on Future Directions in Information Access*, FDIA ’17, 2017.
- [71] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, “Processing social media messages in mass emergency: A survey,” *ACM Comput. Surv.*, vol. 47, pp. 67:1–67:38, June 2015.
- [72] S. González-Bailón, J. Borge-Holthoefer, A. Rivero, and Y. Moreno, “The dynamics of protest recruitment through an online network,” vol. 1, p. 197, 12 2011.
- [73] S. Asur and B. A. Huberman, “Predicting the future with social media,” in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT ’10, (Washington, DC, USA), pp. 492–499, IEEE Computer Society, 2010.
- [74] N. Alsaedi, P. Burnap, and O. Rana, “Can we predict a riot? disruptive event detection using twitter,” *ACM Transactions on Internet Technology*, vol. 17, pp. 18:1–18:26, Mar. 2017.
- [75] H. Becker, M. Naaman, and L. Gravano, “Beyond trending topics: Real-world event identification on twitter,” in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.
- [76] K. N. Vavliakis, A. L. Symeonidis, and P. A. Mitkas, “Event identification in web social media through named entity recognition and topic modeling,” *Data Knowl. Eng.*, vol. 88, pp. 1–24, Nov. 2013.
- [77] E. Benson, A. Haghighi, and R. Barzilay, “Event discovery in social media feeds,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT ’11,

- (Stroudsburg, PA, USA), pp. 389–398, Association for Computational Linguistics, 2011.
- [78] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao, “Twitcident: Fighting fire with information from social web streams,” in *Proceedings of the 21st International Conference on World Wide Web, WWW ’12 Companion*, (New York, NY, USA), pp. 305–308, ACM, 2012.
- [79] K. Watanabe, M. Ochi, M. Okabe, and R. Onai, “Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs,” in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM ’11*, (New York, NY, USA), pp. 2541–2544, ACM, 2011.
- [80] O. Ozdakis, P. Senkul, and H. Oguztuzun, “Semantic expansion of tweet contents for enhanced event detection in twitter,” in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 20–24, Aug 2012.
- [81] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins, “The predictive power of online chatter,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD ’05*, (New York, NY, USA), pp. 78–87, ACM, 2005.
- [82] X. Zhang, H. Fuehres, and P. A. Gloor, “Predicting stock market indicators through twitter i hope it is not as bad as i fear,” *Procedia - Social and Behavioral Sciences*, vol. 26, pp. 55 – 62, 2011. The 2nd Collaborative Innovation Networks Conference - COINs2010.
- [83] R. Bandari, S. Asur, and B. A. Huberman, “The pulse of news in social media: Forecasting popularity,” *CoRR*, vol. abs/1202.0332, 2012.

- [84] K. Radinsky and E. Horvitz, “Mining the web to predict future events,” in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM ’13, (New York, NY, USA), pp. 255–264, ACM, 2013.
- [85] N. Kallus, “Predicting crowd behavior with big public data,” in *Proceedings of the 23rd International Conference on World Wide Web*, WWW ’14 Companion, (New York, NY, USA), pp. 625–630, ACM, 2014.
- [86] S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan, “Planned protest modeling in news and social media,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI’15, pp. 3920–3927, AAAI Press, 2015.
- [87] G. Korkmaz, J. Cadena, C. J. Kuhlman, A. Marathe, A. Vullikanti, and N. Ramakrishnan, “Combining heterogeneous data sources for civil unrest forecasting,” in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 258–265, Aug 2015.
- [88] S. Yıldırım and T. Yıldız, “An unsupervised text normalization architecture for Turkish language,” in *16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*, 2015.
- [89] D. Torunoğlu and G. Eryiğit, “A cascaded approach for social media text normalization of Turkish,” in *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pp. 62–70, 2014.
- [90] D. Yuret and M. De La Maza, “The greedy prepend algorithm for decision list induction,” in *International Symposium on Computer and Information Sciences*, pp. 37–46, Springer, 2006.
- [91] A. A. Akin and M. D. Akin, “Zemberek, an open source nlp framework for Turkic languages,” *Structure*, vol. 10, pp. 1–5, 2007.

- [92] R. Dehkharghani, Y. Saygin, B. Yanikoglu, and K. Oflazer, “Sentiturknet: a Turkish polarity lexicon for sentiment analysis,” *Language Resources and Evaluation*, vol. 50, no. 3, pp. 667–685, 2016.
- [93] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment strength detection in short informal text,” *Journal of the Association for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.
- [94] C. Türkmenoglu and A. C. Tantug, “Sentiment analysis in Turkish media,” in *International Conference on Machine Learning (ICML)*, 2014.
- [95] “Github: Bilkentinformationretrievalgroup/ tubitak215e169,” 2017.
- [96] J. H. McDonald, *Handbook of biological statistics*, vol. 2. Sparky House Publishing Baltimore, MD, 2009.

Appendix A

Details of Discovering Story Chains

A.1 Output Chains

The story chains that are obtained by our framework algorithm, *hZZ*, and annotated in the second user study to compare with baselines are listed in Tables A.1, A.2, and A.3 for the cases *Ukrainian Riots*, *Trucks Going to Syria*, and *Allegations to Fenerbahçe*, respectively.

A.2 User Study for Comparing with Baselines

The details of the user study that we conduct to compare the success of our story-chain discovery method with baseline methods are given in Table A.4 and A.5.

Table A.1: The output of our framework algorithm, for the case *Ukrainian Riots*, to be compared with baseline methods in the second user study.

Date	Snippet
01/25/2014	Ukrayna'da uzlaşma yok!. Ukrayna-AB ile ortaklık anlaşmasının imzalanmamasına iki aydır isyan eden muhalifler...
01/30/2014	"Kamu binalarını derhal boşaltın". Ukrayna Parlamentosu Başkanı Volodimir Rybak, dün kabul edilen af yasasına göre...
02/18/2014	Göstericiler bakanlık binasını ele geçirdi. Ukrayna'nın başkenti Kiev'de bu sabah parlamentoya doğru yürüyüşe geçen...
02/19/2014	Kolluk kuvvetleri meydana girdi. Kiev'deki Bağımsızlık Meydanı'ndaki protestocuları işgal ettikleri yerlerden...
02/19/2014	Dünya ayağa kalktı, Putin sustu!. Ukrayna'da yaşanan bu uzun ve kanlı gecenin bilançosu sabah saatlerinde netleşmeye...
02/19/2014	Ukrayna muhalefetine gözdağı. İlk günden bu yana muhalefeti diyaloga çağırıldığını ve hep uzlaşmadan yana bir tutum...
02/19/2014	Venezuela'da muhalif lider teslim oldu. Dün başkent Caracas'ta düzenlenen gösteriye katılarak kısa bir konuşma yapan...
02/19/2014	Ukrayna'da meydanlar kan gölü!. Ukrayna'nın başkenti Kiev'de operasyon katliama dönüştü. Muhalefetin meclisi içten ve...
03/03/2014	Rusya'dan Ukrayna'ya acil ultimatoma!. Rusya'nın, Ukrayna'daki askeri güçlerin Kırım'ı terk etmesi için Kiev...
03/04/2014	Putin: Ukrayna'da darbe yapıldı, yönetim gayrimeşru. Bu sabah sürpriz bir kararla, Ukrayna sınırında tatbikat yapan...
03/09/2014	Kırım'da Ukrayna yanlılarına meydan dayadı!. Olaylarda aktivistlere kamçı ve golf sopaları ile saldırıldığını...
03/10/2014	Rusya'dan Ukrayna'ya çok sert açıklama!. Bakanlıktan yapılan açıklamada, Sağ sektör (aşırı sağ) örgütünün Ukrayna'nın...

Table A.2: The output of our framework algorithm, for the case *Trucks Going to Syria*, to be compared with baseline methods in the second user study.

Date	Snippet
01/02/2014	Hatay'daki TIR 'devlet sırrı' çıktı. TIR Hatay Kırıkhan'da önceki gece ihbar üzerine durdurulan TIR krize yol açarken...
01/03/2014	Birleşmiş Milletler: TIR'ları arayın!. Hürriyet gazetesinden Razi Canıklıgil'in haberine göre, BM Genel Merkezi'nde...
01/03/2014	CHP'den tır için suç duyurusu. Hatay Milletvekili Refik Eryılmaz, Hatay Cumhuriyet Başsavcılığına, Hatay...
01/04/2014	Vural'dan çok sert açıklamalar. MHP Grup Başkanvekili Oktay Vural, Edirne'de partisinin aday tanıtım toplantısında...
01/08/2014	AKP'li Aktay'dan bomba TIR açıklaması!. CNN Türk'te Dört Bir Taraf programına Nagehan Alçı'nın yokluğunda katılan AKP...
01/12/2014	Pekmez bidonunda esrar. Jandarma, Nevşehir - Aksaray karayolu üzerinde Acıgöl ilçesine bağlı Tepeköy yol ayrımında...

Table A.3: The output of our framework algorithm, for the case *Allegations to Fenerbahçe*, to be compared with baseline methods in the second user study.

Date	Snippet
01/17/2014	Aziz Yıldırım'ın cezası onandı. şike davasında Yargıtay kararını açıkladı: Kısmen onandı, kısmen düştü...
01/18/2014	Yıldırım'dan bomba açıklamalar!. Fenerbahçe Başkanı Aziz Yıldırım, resmi site üzerinden Yargıtay'ın verdiği şike...
01/19/2014	MHP'de Aziz Yıldırım çatlağı!. MHP Genel Başkanı Devlet Bahçeli, Fenerbahçe Başkanı Aziz Yıldırım'ın şike ve teşvik...
01/19/2014	Fenerbahçe taraftarından flaş karar. Yargıtay kararından sonra taraftarlar bir kez daha sokaklara dökülüyor...
01/21/2014	Aziz Yıldırım İstanbul'a döndü. Aziz Yıldırım'ı taşıyan uçak saat 21.18'de Sabiha Gökçen Havalimanı'na indi...
01/21/2014	Aziz Yıldırım'ın uçağı indi. Fenerbahçe'nin taraftar grupları Aziz Yıldırım'ın havalimanında taraftarlar tarafından...
01/22/2014	Zekeriya öz'e avukat şoku!. Fenerbahçe Başkanı Aziz Yıldırım'ın, Fenerbahçe Yüksek Divan Kurulu toplantısında...
01/23/2014	Çocuk gelin cinayetinde karar. Aile içi şiddetten kaçarak sığındığı baba evinde 17 yaşındaki Emine Yağla'yı öldüren...
01/25/2014	Savcı Aytaç Durak'a beraat istedi. Adana özel Yetkili, 8. Ağır Ceza Mahkemesindeki, 73 sanıklı çete davasında Savcı...
01/29/2014	3 bin TL'lik yumurta. Mahkeme Ali Sürmeli'nin, suç riskinin azaltılması, toplumun yasalara bağlı bir ferdi olmaya...
01/31/2014	UEFA Yargıtay kararını istedi!. Fenerbahçe'ye TFF'nin seyircisiz oynama cezası vermesinin ardından bir şok da...
02/04/2014	Aziz Yıldırım'a Sivas şoku!. Sivasspor ile Fenerbahçe arasında pazar günü oynanacak maçta konuk ekibin...
02/06/2014	Tahkim'den Fenerbahçe'ye müjde. Sarı-Lacivertliler'in 1 maç seyircisiz oynama cezası kaldırıldı...
03/05/2014	Aziz Yıldırım'ın avukatlarından flaş açıklama. Fenerbahçe Kulübü, Yargıtay Cumhuriyet Başsavcılığı'ndaki ceza...
03/10/2014	Utancı gecesinde kareler!. çıkan olaylar nedeniyle Trabzonspor - Fenerbahçe maçı, hakem Bülent Yıldırım tarafından...

Table A.4: The details of the user study that we conduct to compare the success of our story-chain discovery method with baseline methods, in terms of *relevance* and *coverage*. *hZZ*: Hybrid and Zigzagged Search, with three baselines, *sTDT*: Simple TDT, *aTDT*: Adaptive TDT, *GN*: Google News.

	Username	Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe						
		sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ	
Relevance	akin	2	-1	-1	2			2	1	1	-2	-2	1	1
	arslan	1	-1	2	2			1	1	1	-1	-1	1	1
	arslan	2	-1	-2	1			1	1	1	-2	-2	1	1
	bahceci	2	-2	1	2			2	1	2	-1	-1	1	1
	bostanci	1	-1	-1	1			-1	1	1	-2	-1	1	1
	demirtas	2	-1	1	2			1	1	1	-2	-2	1	1
	fazlican	2	-1	2	2			1	-1	1	-2	2	-1	1
	inel	1	-1	2	2			-1	1	1	-1	-1	2	1
	sav	-1	-2	-1	1			1	-1	1	-1	-1	1	1
	sumbul	1	-1	1	1			1	1	1	-1	-1	1	1
Coverage	torun	1	-1	-1	2			1	1	1	-1	1	1	1
	yildiz	1	-2	1	2			1	1	1	-2	-1	1	1
	akin	2	-1	1	1			1	1	1	-2	-2	-1	-1
	arslan	1	1	2	2			2	2	2	-1	-1	1	1
	arslan	2	-2	-2	2			2	2	1	-2	-1	1	1
	bahceci	1	-2	1	2			1	1	2	-1	-1	1	1
	bostanci	1	-1	-1	1			-1	1	1	-2	-1	-1	-1
	demirtas	1	-1	2	2			1	1	1	-2	-2	1	1
	fazlican	2	1	2	2			-1	-1	1	-2	-1	-1	1
	inel	1	1	2	2			1	2	1	-1	-1	2	1
sav	2	1	1	1			1	1	1	1	-1	1	1	
sumbul	1	1	1	1			1	1	1	1	1	1	1	
torun	2	-1	-1	2			1	1	1	1	1	2	1	
yildiz	1	-1	2	2			-2	1	2	-2	-1	1	1	

Table A.5: The details of the user study that we conduct to compare the success of our story-chain discovery method with baseline methods, in terms of *coherence* and *ability to disclose relations*. *hZZ*: Hybrid and Zigzagged Search, with three baselines, *sTDT*: Simple TDT, *aTDT*: Adaptive TDT, *GN*: Google News.

	Username	Ukrainian Riots			Trucks Going to Syria			Allegations to Fenerbahçe					
		sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ	sTDT	aTDT	GN	hZZ
Coherence	akin	1	-2	-2	-1	-2	2	-1	1	-2	-2	-1	-1
	arslan	1	1	1	2	-1	-1	1	-1	-2	-1	-1	-1
	aslan	1	-1	-2	2	2	2	1	2	-2	-1	1	1
	bahceci	-1	-1	1	2	-1	-1	-1	-1	-1	-1	1	1
	bostanci	-1	-2	1	1	-2	-1	-1	-1	-2	-1	-1	-1
	demirtas	1	-2	1	2	-2	-1	-1	1	-2	-2	-1	-1
	fazlican	2	-1	-2	2	-2	2	2	1	-2	2	2	1
	inel	-1	-1	1	1	-2	-1	1	1	-1	-1	1	1
	sav	1	-2	1	1	-1	1	1	-2	-2	-1	-1	-1
	sumbul	-1	-1	1	1	-1	-1	-2	-1	-2	-2	-1	-2
	torun	-1	-1	-1	1	-1	2	2	-1	1	-1	1	2
	yildiz	-1	-2	1	2	-2	1	1	1	-2	-1	-1	1
Disclose Relations	akin	1	1	1	1	-1	1	-1	1	-1	-1	-1	1
	arslan	1	1	1	1	-1	1	1	1	1	1	1	1
	aslan	1	1	1	1	1	1	1	1	-1	-1	1	1
	bahceci	1	-1	1	1	1	1	1	1	1	1	1	1
	bostanci	1	-1	1	-1	-1	-1	1	1	-1	-1	1	-1
	demirtas	-1	-1	1	1	-1	-1	1	-1	-1	-1	1	-1
	fazlican	1	1	1	1	-1	1	1	1	-1	1	1	1
	inel	1	-1	1	1	-1	-1	1	1	-1	-1	1	1
	sav	1	1	1	1	1	1	1	1	-1	-1	1	1
	sumbul	1	1	1	1	1	1	1	1	1	1	1	1
	torun	1	-1	-1	1	1	1	1	1	1	1	1	1
	yildiz	1	-1	1	1	-1	1	1	1	-1	1	1	1

Appendix B

Details of Selecting Public Front-pages

B.1 Main User Study

The details of the main user study that we conduct to evaluate the success of our front-page news selection method are given in Table B.1.

B.2 Additional User Study

The details of the additional user study to support the success of our front-page news selection method are as follows. The total number of news articles is 15,844. The number of annotators is 4, who are undergraduate students. Our methodology is that all annotators read all news articles and have to choose one of four importance degree scores: 1 (not important at all), 2 (not important), 3 (important), and 4 (exactly

important). Since the average of annotation scores for a news article is continuous, we conclude that the average scores that are higher than 2.5, imply important news articles, and those that are lower than 2.5 are for unimportant news. The number of news articles that have scores higher than 2.5 are 1,315, and lower than 2.5 are 14,529.

Table B.1: The details of annotation results of our front-page news selection method.

Username	Calculated Time (min)	Declared Time (min)	Dataset Importance	Dataset Diversity
ayasar	259.90	180.00	0.48	0.83
cagri	194.60	147.00	0.62	0.86
devrim	58.80	85.00	0.54	0.89
dilan	182.10	114.00	0.53	0.87
fcan	4618.50	152.00	0.62	0.88
gece	111.00	240.00	0.65	0.68
gunduz	66.20	179.00	0.46	0.52
hamed	591.60	252.00	0.62	0.51
hayri	145.60	130.00	0.70	0.90
mcan	148.30	145.00	0.41	0.56
memre	67.60	95.00	0.51	0.66
mustafa	77.10	101.00	0.46	0.68
nesli	110.70	181.00	0.38	0.73
sarp	74.90	85.00	0.27	0.53
semih	55.30	62.00	0.59	0.71
sermet	55.40	77.00	0.51	0.94
tolgac	120.60	92.00	0.57	0.91
tolgay	103.10	96.00	0.48	0.86
tom	120.20	115.00	0.50	0.84
Avg.	376.92	133.05	0.52	0.76
Min.	55.30	62.00	0.27	0.51
Max.	4618.50	252.00	0.70	0.94
Std.Dev.	1034.33	53.72	0.10	0.15
Median	111.00	115.00	0.51	0.83

Appendix C

Details of Filtering Microblogs

C.1 Collection

A sample instance from BilPredict-2017 is given in Figure C.1. We also list 10 of tweets that are filtered for the event titled “Aziz Sancar won the Nobel Prize in Chemistry 2015”, in Table C.1.

```
<INSTANCE>
<ID>17</ID>
<EVENT>
<NAME_TR>AZİZ SANCAR NOBEL KİMYA ÖDÜLÜ'NÜ KAZANDI</NAME_TR>
<NAME_EN>AZİZ SANCAR WINS NOBEL PRIZE FOR CHEMISTRY </NAME_EN>
<PLACE>İSVEÇ</PLACE>
<LINK>
<URL>http://www.hurriyet.com.tr/nobel-kimya-odulunu-turk-asilli-aziz-sancar-kazandi-aziz-sancar-kimdir-30255503</URL>
<DATE>
<YEAR>2015</YEAR>
<MONTH>10</MONTH>
<DAY>7</DAY>
<TIME>13:28</TIME>
</DATE>
</LINK>
</EVENT>
<REACTION>
<CATEGORY_TR>SOSYAL_OLUMLU</CATEGORY_TR>
<CATEGORY_EN>SOCIAL_POSITIVE</CATEGORY_EN>
```

Figure C.1: A sample instance from BilPredict-2017.

Table C.1: Sample filtered tweets for the instance from 2015 given in Figure C.1.

Date	Tweet	Sentiment Score
Oct 07 15:13:56	@AzzSancar hocam sözleriniz bizi aydınlatacak...bu sözünü ilke edineceğim ve bende nobel alacağım.. Tebrikler....	2.0
Oct 07 15:21:31	Nihayet bir Türk Doktor nobel ödülü aldı! Tebrikler Aziz Sancar. #NobelPrize Gurur duyuyoruz...	3.0
Oct 07 21:35:28	Aziz Sancar , nobel ödülü alarak göhsümüzü kabarttı gel gelelim yarın tv de siyasetçilerin aptalca atışmaları izliyez arada kaynıyıp gitcek	-3.0
Oct 07 21:47:07	@hyahya_kurdi Aziz Sancar Hocamızı aldığı nobel kimya ödülünden dolayı tebrik ediyoruz	2.0
Oct 07 23:04:12	Mardinin savur ilçesinin bir köyünden bir çocuk çıkacak .gidip dünyada bilim kimya nobel ödülü alacak . helal olsun #AzizSancar	2.0
Oct 08 00:27:59	ülkeye bak be! Adam nobel ödülü almış tebrik yok hangi partiden hangi ırktan sualleri var! Bekleme yapmayalım bence.. #AzizSancar	2.0
Oct 08 08:36:53	PROF dr. Aziz Sancar nobel ödülünü aldı istersen başarı oluyor tebrikler	2.0
Oct 08 09:06:25	Adam nobel ödülü almış,ne için aldığından çok etnik kökeni tartışılıyor. Böyle *** bi *** yaramaz toplumuz işte...	-2.0
Oct 08 10:54:12	nobel kimya ödülünü alan Aziz Sancar'a tebrikler..gurur duyduk	2.0
Oct 08 14:37:06	ülkesine ve tarihine sövmeden nobel alınabiliyormuş bunu öğrendik cânı gönülden tebrik ediyoruz	2.0