TOPIC IDENTIFICATION WITHIN MICROBLOG POST COLLECTIONS

by

Ahmet Yıldırım

B.S., Karadeniz Technical University, 2005

M.S., Software Engineering, Boğaziçi University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2017

ACKNOWLEDGEMENTS

I want to thank my thesis advisor Dr. Suzan Üsküdarlı for her patience and invaluable support through this dissertation. You were always there with your full support, insight, and encouragement. You were tolerant, and kind. You saw challenging conditions in my life, not only were you there and supporting, but also you encouraged me to continue. You encouraged me to live the most cheerful ways in positive situations. You offered me a wonderful blend of autonomy and guidance throughout my time as a doctoral student. I appreciate your continuous support. This thesis would never finalize without your guidance and support.

I also want to thank my co-advisor, Prof. Haluk Bingöl for his suggestions on my dissertation. I appreciate your contributions and kindness. You were supportive in all conditions. Your support helped me a lot. Working with you was a privilege to me.

I extend my deep gratitude to the members of my dissertation progress committee, Dr. Suzan Üsküdarlı, Prof. Haluk Bingöl, Assoc. Prof. Arzucan Özgür and Prof. Yağmur Denizhan. Thanks for the time and expertise. I appreciate all the suggestions that helped my work to evolve.

I wish to thank TAM Project, especially Prof. Ufuk Çağlayan. I appreciate his support and trust. I thank TETAM, CMPE, SOSLAB and their members for supplying a great place to study. It is always a pleasure to work with you. I thank Prof. Cem Ersoy for his support in my study.

I thank Netaş for the scholarship support throughout this dissertation.

I thank Suzan Üsküdarlı, Arzucan Öztürk, Haluk Bingöl, Yağmur Denizhan, Nuri Taşdemir, Onur Güngör, Uzay Çetin, T.B. Dinesh, Dağhan Dinç, Amaç Herdağdelen, Çağıl Uluşahin, Birkan Yılmaz, Erdem Beğenilmiş, Fulya Sarı, Nadin Kökciyan, Mehmet Şükrü Kuran, Serhan Daniş, Samet Atdağ, Melih Barsbey, Halid Can Yıldırım, Selcan

Çınar Yıldırım, Mehmet Gökhan Habiboğlu who I had discussions with about this work and other research related topics. Spending time with you was really a great opportunity for me to extend my thoughts.

I especaily thank Nuri Taşdemir for his support, kindness, and patience throughout this study. You were always there, supporting me in any way required. Again, thank you!

I thank ÇOT! members Çetin Meriçli, Onur Dikmen, Tekin Meriçli, İsmail Arı, Yunus Durmuş, Gaye Genç, Yunus Emre Kara, and Serhan Daniş. I had a great time with you. I can't imagine what my PhD would look like, when I turn back and think about it years later, without the time I spent with you. I will be waiting for the re-union and the "artist breakfasts".

I thank my brother Halid Can Yıldırım, my mother Hatice Yıldırım and my father Yasin Yıldırım. You have always supported me. Without your support and love I would have never done this.

I thank my friends who are as close as family: Nuri Teşdemir, Bülent Hoca, Yeliz Polat Hoca, Erdem Beğenilmiş, and Gamze Pat Beğenilmiş. Spending time, sharing life and discussions with you about anything has been a great enjoyment for me.

Finally, I thank my lovely wife Selcan Çınar Yıldırım, for her continuous support over the years. I love you and am so glad I have you. Your support finalized this work. You are one of the two great things in my life. And about the other thing... my son, Ervin. It has been four months since your arrival. How nice that you did arrive. You cheered my last months while I was busy with this thesis. You are so little, so fragile. How nice to love you. I hope you grow up, and one day contribute to the knowledge of humanity.

ABSTRACT

TOPIC IDENTIFICATION WITHIN MICROBLOG POST COLLECTIONS

This thesis aims to identify topics in collections of microblog posts, where topics correspond to a set of related topic elements. The first approach, Boun-TI, examines the use of Wikipedia – well written cross-domain articles – to capture topics within microblog posts that are messy, unstructured, and fragmented. The topic elements are identified based on their *tf-idf* scores, where the microblog post set is considered as a single document for *tf* computation. For *idf* computation, a public stream post set is used where each post is considered as a document. The *tf-idf* vectors of Wikipedia articles are computed, and the cosine similarity of the *tf-idf* vectors determine the topics. This approach was evaluated with more than 1 million tweets gathered during the 2012 US presidential election, resulting in a precision of 0.96 and $F_1 = 1$.

The second approach, S-Boun-TI, examines the generation of semantically structured topics, so that they can be further processed to yield more information. S-Boun-TI considers distinguishing elements of a post set as linked entities. Co-occurrence of two elements in the same post is considered as a relation. The related element sets which form topics are maximal cliques of the graph of elements and relations. To express topics, an ontology for microblog topics is introduced. The topics can be utilized in conjunction with LOD. Over 1M posts during the 2016 U.S. presidential election debates, and other events such as the death of Carrie Fisher and the Dakota Access Pipeline demonstrations were considered for evaluation. Quantitative and qualitative observations are provided and example SPARQL queries and their results are presented to show the utilization of the topics. Both approaches gave promising results and are suitable for future research and development. S-Boun-TI has been found to represent related elements better then Boun-TI.

ÖZET

KISA İLETİ KÜMELERİNDE KONU ALGILAMA

Bu tez, konuların bir dizi ilgili unsura karşılık geldiği kısa ileti mesaj kümelerindeki konuları çıkarmayı amaçlamaktadır. İlk yaklaşım olan BOUN-TI, dağınık, yapılandırılmamış ve parçalanmış kısa iletilerin içindeki konuları yakalamak için, herhangi bir alana özel olmayan daha düzgün yazılmış olan Wikipedia'nın kullanımını inceler. Konu unsurlarını bulmak için kullanılan *tf* hesaplamasında kısa ileti mesaj kümelerini tek bir belge olarak kabul eder. Başka bir genel kısa ileti kümesi, *idf* hasaplamada kullanılır ve bu hesaplamada her bir kısa iletiyi bir belge olarak kabul eder. İngilizce Wikipedia makalelerinin *tf-idf* vektörlerini hesaplar. *tf-idf* vektörlerinin kosinüs benzerliği konuları belirler. Bu yaklaşım 2012 ABD Seçimi sırasında toplanan 1 milyonun üzerinde mesaj ile değerlendirildi ve sonuç olarak $0,96$ hassaslık skoru elde edildi ($F_1 = 1$).

İkinci yaklaşım olan S-BOUN-TI, anlamsal olarak yapılandırılmış konuların üretilmesini inceler ve bu sayede, daha fazla bilgi elde etmek için işlenebilmelerini sağlar. S-BOUN-TI, bir mesajın elemanlarını bağlantılı parçalar olarak kabul eder. Aynı mesajda iki parçanın birlikte olmasını bir ilişki olarak kabul eder. İlgili elemanlar ve aralarındaki ilişkilerin çizgesinden, en büyük klikleri kullanarak konuları belirler. Konuları ifade etmek için bu tezde tanımlanan $Topic_{\mathcal{O}}$ ontolojisini kullanır. Konu elemanıları Bağlı Açık Verilerdeki (LOD) kaynaklara bağlı olduğu için, LOD ile birlikte kullanılabilirler. Bu yaklaşımı incelemek için 2016'daki ABD seçimleriyle ilgili tartışmalar süresince, Carrie Fisher'ın ölümü ve Kuzey Dakota'daki boru hattı gösterileri gibi diğer olaylarda atılan 1 milyondan fazla kısa ileti değerlendirmeye alınmıştır. Nicel ve nitel gözlemler ve konuların kullanımını göstermek örnek için SPARQL sorguları ve sonuçları sunulur. Her iki yaklaşım umut verici sonuçlar vermiştir ve gelecekteki araştırma ve geliştirme için uygundur. S-BOUN-TI'nin ilgili elemanları BOUN-TI'den daha iyi temsil ettiği görülmüştür.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES x
LIST OF TABLES xiv
LIST OF SYMBOLS xvi
LIST OF ACRONYMS/ABBREVIATIONS xviii
1. INTRODUCTION 1
2. LITERATURE REVIEW 5
2.1. Approaches that provide words or phrases as topics 9
2.2. Approaches that manually define topics 12
2.3. Approaches that provide representative posts as topics 13
2.4. Approaches that provide summarization phrases as topics 14
2.5. Approaches that classify posts to extract domain specific information . 15
2.6. Conclusions 16
3. BACKGROUND 18
3.1. About microblog posts 18
3.2. Twitter API 18
3.3. Wikipedia 19
3.4. Vector space model and document similarity 19
3.5. Entity Linking 20
3.6. Linked data 20
3.7. W3C Vocabularies and Ontologies 21
3.8. Ontology, vocabulary and data namespace prefixes 22
3.9. Tools 22
4. TOPIC IDENTIFICATION USING WIKIPEDIA 24
4.1. Boun-TI: Topic Identification Using Wikipedia 25
4.2. Prototype 31
4.3. Experiments and Results 35

4.3.1. Datasets . . . 36
4.3.2. Experiments . . . 37
4.3.3. Evaluation results . . . 38
4.4. Examining Topics over Time . . . 40
4.5. Identifying topics for single versus multiple microblog posts . . . 44
4.6. Conclusions . . . 47
5. TOPICO: AN ONTOLOGY FOR MICROBLOG TOPICS . . . 49
5.1. Agents . . . 50
5.2. Locations . . . 51
5.3. Temporal expressions . . . 52
5.4. Meta information . . . 54
5.5. Object and data properties . . . 56
6. EXTRACTING MACHINE INTERPRETABLE TOPICS . . . 58
6.1. S-Boun-TI: Structured Topic Identification . . . 61
6.1.1. Candidate element extraction . . . 62
6.1.1.1. Agent identification . . . 64
6.1.1.2. Location identification . . . 65
6.1.2. Candidate element improvement . . . 67
6.1.3. Relating elements . . . 68
6.1.4. Identifying topics . . . 68
6.2. Prototype . . . 69
6.2.1. Candidate element identification . . . 71
6.2.1.1. Person identification . . . 72
6.2.1.2. Location identification . . . 73
6.2.1.3. Temporal expression identification . . . 74
6.2.2. Relating elements . . . 74
6.2.3. Topic identification . . . 74
6.2.4. Semantic topic instantiation . . . 76
6.3. Experiments and results . . . 76
6.3.1. Post set characteristics . . . 76
6.3.2. Entity linking . . . 78

6.3.3. Co-occurrence graphs and cliques 81
6.3.4. About resulting topics 84
6.3.5. Processing Topics 88
6.4. Observations 95
6.5. Conclusions 100
7. RELATED WORK AND COMPARISON 102
7.1. Related work of Boun-TI 102
7.2. Related work of S-Boun-TI 104
7.3. Comparison with LDA 105
7.3.1. Comparing Boun-TI and LDA topics 107
7.3.2. Comparing S-Boun-TI and LDA topics 111
7.4. Comparing Boun-TI and S-Boun-TI topics 114
7.5. Conclusions 115
8. DISCUSSION AND FUTURE WORK 122
9. CONCLUSION 124
REFERENCES 126
APPENDIX A: ADDITIONAL STOPWORDS FOR TWITTER POST-PROCESSING IN BOUN-TI 141
APPENDIX B: SBOUN-TI TEMPORAL EXPRESSION IDENTIFIERS . . . 143
APPENDIX C: ENTITY LINK IMPROVEMENTS 145
APPENDIX D: RESULT OF QUERIES IN FIGURE 6.14 146

LIST OF FIGURES

Figure 2.1. Sample assignments of documents, LDA topics and words 10

Figure 2.2. An example graph for summarization that can be obtained by the approach of Sharifi *et al.* [1] 14

Figure 3.1. Six spots and their associated entities as suggested by TagMe for a given tweet text 21

Figure 4.1. Overview of the process of topic extraction from a set of microblog posts 26

Figure 4.2. BOUN-TI algorithm 32

Figure 4.3. The BOUN-TI evaluation interface 38

Figure 4.4. The heatmap of topics identified during the 2012 US election debates 41

Figure 4.5. The "Big Bird" and "Christianity and Abortion" topics and their scores during different the first PD and the Vice PD 43

Figure 5.1. Class hierarchy of $Topic_{\mathcal{O}}$. 50

Figure 5.2. Object properties of $Topic_{\mathcal{O}}$ 51

Figure 5.3. A "topico:Topic" instance regarding Hillary Clinton, Donald Trump and Racism 52

Figure 5.4. An example of a topic's "topico:isAbout" property. The topic is about Stop and frisk in New York 53

Figure 5.5. A topic instance with an organization (United States Department of Justice) and a person (Donald Trump) 53

Figure 5.6. An example topic with a location (North Dakota) 54

Figure 5.7. A topic with an example temporal expression "Now" 55

Figure 5.8. Example of two temporal expressions "2016" and "Tonight" 55

Figure 5.9. An example topic with expressing meta information such as the observation interval, creation time, and the maker of the topic . . 56

Figure 5.10. Inverse properties of the object properties of $Topic_{\mathcal{O}}$. 57

Figure 6.1. Several posts that some elements are related 58

Figure 6.2. Illustration of topic identification 60

Figure 6.3. Overview of identifying semantic topics from a set of microblogs . 62

Figure 6.4. Semantic topic extraction algorithm 63

Figure 6.5. Example to topics extracted from a sample graph 69

Figure 6.6. Overview of topic extraction from a set of microblog posts 70

Figure 6.7. Weight of edges and the percentage of those edges in the post sets 83

Figure 6.8. Various cliques and their amounts before and after post processing cliques 85

Figure 6.9. The interface used to view and inspect semantic topics 88

Figure 6.10. The query for finding the people who most often appeared in the same topic with Hillary Clinton. 89

Figure 6.11. The query for determining when the topics related to "Women's Issues" were talked about 89

Figure 6.12. A query for determining when the topmost 50 topics related to Hillary Clinton and Donald Trump emerged 91

Figure 6.13. Various issues discussed during the four 2016 US presidential election debates 92

Figure 6.14. Query of politicians in the topics applied on three endpoints: S-Boun-TI-Fuseki, DBpedia, and DBpedia-Wikidata 94

Figure 6.15. Query of rock musicians, and locations of concerts 95

Figure 7.1. Comparison algorithm of Boun-TI and LDA topics 108

Figure 7.2. The histogram of all comparison scores of Boun-TI and LDA topics 110

Figure 7.3. Heatmap of comparison scores of Boun-TI and LDA topics over the four debates 118

Figure 7.4. Comparison algorithm of S-Boun-TI and LDA topics 119

Figure 7.5. The histogram of all comparison scores of S-Boun-TI and LDA topics 120

Figure 7.6. Heatmap of comparison scores of S-Boun-TI and LDA topics over the four debates 121

LIST OF TABLES

Table 3.1. The ontologies and data name spaces referred to in this thesis . . . 23

Table 4.1. Characteristics of presidential debates (PD) and the vice PD post sets . . . 36

Table 4.2. Randomly selected time intervals of the debates . . . 39

Table 4.3. The precision scores of the top 1, 5, and 10 topics according to μ . 39

Table 4.4. The F_1 measure for the inter-annotator agreement 1^{st}, 5^{th}, and 10^{th} topmost topics according to μ . . . 40

Table 4.5. Comparison of BOUN-TI topics and topics from aggregating TagMe linkings . . . 45

Table 4.6. Comparison of BOUN-TI topics and topics from aggregating TagMe linkings . . . 45

Table 4.7. Topics obtained by aggregating TagMe linkings that are extracted from [28,30) minutes of the first presidential debate, and [80,82) minutes of the vice presidential debate . . . 46

Table 6.1. Characteristics of post sets collected from Twitter . . . 77

Table 6.2. The types of entities in the post sets . . . 79

Table 6.3. The number of vertices and edges before and after pruning entity co-occurrence graphs . . . 82

Table 6.4. The percentage of tweets that contribute to the vertices, edges and topics. 84

Table 6.5. Characteristics of topics according to the number of persons, locations and temporal expressions 86

Table 7.1. Comparison of BOUN-TI topics with LDA topics 109

Table 7.2. Comparison of S-BOUN-TI topics with LDA topics 113

Table A.1. Stopwords used in processing tweets 141

Table B.1. The enitities linked to temporal expression spots 143

Table C.1. Some spots and their corresponding initial and relinked entities . . 145

Table D.1. Result of queries in Figure 6.14 which are the politicians existing in topics 146

LIST OF SYMBOLS

a	The number of items that both annotators give positive annotation in inter annotator agreement calculation
b	The number of items that the first annotator gives positive while the second annotator gives negative annotation in inter annotator agreement calculation
c	The number of items that the second annotator gives positive while the first annotator gives negative annotation in inter annotator agreement calculation
d	The number of items that both annotators give negative annotation in inter annotator agreement calculation *or* a document
C	The set of all possible token-*tf-idf* value pairs (t, n) of topics
idf	Inverse document frequency function
F	A set of token-*tf-idf* value pairs (t, r) in the Algorithm 4.2
F_1	Inter-annotator aggreement rate
d	A set of words which represents a document that is also a post text
m_1	First minute of a time interval
m_2	Last minute of a time interval
MB_{tf}	A set of token-*tf* value pairs (t, n) of input posts
$\mathbb{N}$	The set of natural numbers
n	*tf* value
Pub	Public Stream Post set
Pub_{idf}	A set of token-*idf* value pairs (t, r) of input posts
p	*idf* value *or* a variable in computations
p_i	A microblog post
$\mathbb{R}_{\geq 0}$	The set of non-negative real numbers
r	*idf* value in formal representations *or* the variable that is set to *tf-idf* values of microblog post tokens
S	Set of all strings
t_i	An S-Boun-TI topic

W_{tfidf}	The set of "set of all topics where each element is token-*tf-idf* value pairs (t, n)"
w	A word or token
x	A variable in algorithm in Figure 4.2
y	A variable in algorithm in Figure 4.2
Top	The set of BOUN-TI topics and their scores
z	A variable in algorithm in Figure 4.2
α	The function that computes *tf-idf* values of the given microblog post set using sets A and B
β	The function that assigns zero as value to token adds this assignment ($(t, 0)$ pair) to one set if the other does not have that token
$\delta_{a,b}$	Kronecker delta function that returns 1 if a and b are equal, 0 otherwise
Γ	The function that computes cosine similarity of vectors where vectors are represented as sets of token-*tf-idf* (t, p) pairs
μ	Number of top tokens according to their *tf-idf* values for querying in Solr
ϕ	The number of BOUN-TI topics considered for precision and F_1 measure calculations
τ_c	Two clique merging Jaccard similarity threshold
τ_e	Edge removal threshold
$\tau_{e_{min}}$	Edge weight threshold to merge two cliques
τ_{loc}	The post ratio among all posts threshold for an enity to be considered as location
τ_p	Entity spot confidence threshold
τ_{two}	Entity frequency ratio that is checked against frequencies of entities in 2-cliques to decide removal of the clique
τ_ρ	Entity link confidence threshold

LIST OF ACRONYMS/ABBREVIATIONS

API	Application Programming Interface
BOUN-TI	Boğaziçi University Topic Identification
C-SPARQL	Continious SPARQL
DCMI	Dublin Core Metadata Initiative
FOAF	Friend of a Friend
GMT	Greenwich Mean Time
JSON	Javascript Object Notation
NLP	Natural Language Processing
OWL	Web Ontology Language
PBS	Public Broadcasting Service
PHP	PHP: Hypertext Preprocessor
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
S-BOUN-TI	Semantic Boun-TI
SPARQL	SPARQL Protocol and RDF Query Language
SIOC	Semantically Interconnected Online Communities
SKOS	Simple Knowledge Organization System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

1. INTRODUCTION

Microblogs allow users to share short texts about anything. One of the most popular microblog systems is Twitter [2] with over 58M posts/day (tweets/day) [3] posted by 340 million active users. Twitter has been prominent during many significant events: disasters, elections, entertainment events, news and much more. This prominence makes identifying the topics of discussion on Twitter very lucrative, but the number of tweets as well as the characteristics of individual tweets (messy, noisy, unstructured, carelessly written, containing special jargon [4]) makes this difficult. Rich content created in real time, in large quantities, requires automatic methods to extract information from contributions.

Collections of posts can be useful in identifying public opinions, perceptions, emotions and more. This information can be utilized in numerous ways, such as news generation [5], policy making [6], campaign management [7], marketing [8], and disaster management [9]. The high volume of posts, their limited size, and their characteristics require special processing methods. Essential characteristics of microblog posts can be found in Section 3.1.

Topic identification of microblog posts has been an area of interest for researchers. So far, approaches have been applied on a single post [10–15] or a set of posts [16–22]. Single post applications take single posts into consideration whereas other approaches focus on post sets. Some of these approaches utilize external resources such as Wikipedia or DBpedia [11–13, 23–29] in topic identification. Some of them express topics as word or phrase sets [16–22,30–34], indicative posts [18,20,21,23,29,35–37], and semi-human readable phrases [1]. While some approaches aggregate outputs of single post operations to obtain set level outputs [6, 9, 38, 39], others use relationships among words in posts, such as co-occurrence in the same post, probabilistic topic model approaches such as Latent Dirichlet Allocation (LDA) and the position of one word to another [1, 30–35, 37]. Among these approaches, the domain specific ones [6, 9, 15, 23, 38–41] define special keywords and regular expressions, search matches of these definitions in posts, or train a classifier to identify if a

post gives specific information (such as occurrence of an earthquake, or the sickness of the user such as a headache).

The limitation of single post processing is the fact that it limits the context of topic identification. Post set approaches, though they take larger contexts into consideration, only provide word sets, representative posts, and semi-human readable phrases which require further analysis. In response to these limitations, this thesis introduces two methods to produce human readable and machine interpretable topics which do not require further analysis. Both approaches are collective processing approaches where the source is a set of posts, and each post is related to one or more topics, or to no particular topic at all. Furthermore, aspects of a topic are likely distributed over different posts, thus aggregation is considered. The two approaches identify distinctive elements of microblog post sets and relate them to identify topics, and they utilize external resources in the identification process. Using these approaches provides us topics together with their defining elements in a single phrase or machine interpretable structure.

The first method, BOUN-TI, firstly identifies distinctive words which are assumed to be elements of topics. These elements are related using Wikipedia pages. The post set is represented as a vector of *tf-idf* values. The words and their *tf-idf* values are computed using a slightly different method which compiles *idf* of words from another post set.

Using these words, BOUN-TI finds relevant Wikipedia articles using a cosine similarity vector space based approach. The Wikipedia articles are chosen as topic sources since they span across all human interests, include user-generated content and are updated with constant contributions of new material [12, 14]. It is important that Wikipedia is being updated, since microblogs tend to be dominated with content that is temporarily relevant, often fading in importance in a single day. Unlike conventional encyclopedias, Wikipedia contains more topical articles like ("Death of Michael Jackson" or "Emailgate"). It is also interesting to inspect the similarity of contents between posts that are noisy, cryptic and streaming very fast and well-documented articles.

This approach is used on datasets fetched during the heavily covered presidential debates (three debates of the presidential candidates and one debate of the vice presidential

candidates) during the 2012 United States elections. The topics are evaluated, and the evaluation resulted in 0.96 precision with $F_1 = 1$. The results suggest that the proposed approach is promising in identifying topics. Sections 4.1 and 4.3 give details about the approach, experiments and results. Interesting topics were identified which would have been difficult to identify using single post processing approaches rather than post sets. Section 4.5 gives examples of such topics.

The second method, S-BOUN-TI, identifies elements using entity linking, relates them, and identifies topics using the relations. For this approach, the entity linking and relation of these entities create the context of topics. The approach outputs topics in a structured form. The elements of topics are linked to resources in Linked Open Data (LOD). Linked Data is utilized during the processing of posts to identify relevant parts, with special emphasis on the persons, locations, and temporal aspects of topics. The related elements that will be output as topics are identified by processing the output of the maximal cliques algorithm. Additionally, S-BOUN-TI represents the topic structures as Semantic Web resources. Representing topics as the Semantic Web resources enables querying elements in topics, querying elements among topics using their relationships, and querying topics in conjunction with the data in LOD which is beyond what is provided by the microblogging platforms. For topic expression, an ontology called $Topic_{\mathcal{O}}$ is introduced, which is designed to represent microblog topics. The ontological representation of the topics makes them interpretable by machines. Datasets were collected during the 2016 United States presidential election debates and about other various subjects, evaluated, and the resulting topics made available on the web [42]. The topics that are evaluated are relevant and accurate. Examples of the utilization of these topics are presented in Section 6.3.5. Shortcomings of the approach are related to ambiguities and the insufficiency of external resources.

To the best of the author's knowledge, BOUN-TI and S-BOUN-TI are the first studies that extract topics from microblog post sets, using Wikipedia to output a human-readable phrase as a topic, and form a structure to express the elements of a topic in the Semantic Web using an ontology.

The main contributions of this thesis can be summarized as:

- An approach, BOUN-TI, to identify domain-independent topics as Wikipedia articles for sets of tweets,
- An approach, S-BOUN-TI, to extract semantically structured topics from entities that are extracted from posts and linked to Semantic Web resources,
- The development of an ontology to semantically represent topics ($Topic_{\mathcal{O}}$),
- Resources
  - (i) Manually annotated sets of tweets with corresponding topics (30 sets of 6000 tweets each) that were collected during the 2012 United States presidential election debates (post identifiers are provided)
  - (ii) Posts that have been used in evaluations collected during the 2016 United States presidential election debates and about other various subjects (post identifiers are provided)
  - (iii) Inverse document frequency (idf) scores of a public stream over 5 days, which is useful in assessing the values of tokens on Twitter
  - (iv) Term frequency (tf) and idf scores of words in a Wikipedia snapshot
  - (v) Prototypes of the proposed approaches
  - (vi) The topics extracted from the post sets
  - (vii) Codes that have been implemented

The remainder of this thesis is structured as follows: Chapter 2 presents a review of the literature related to extracting what microblog users are talking about. Chapter 3 provides background information about several approaches, services and tools related to this thesis. Chapter 4 introduces the approach, BOUN-TI, to identify topics of microblog post sets using Wikipedia. Chapter 5 introduces the $Topic_{\mathcal{O}}$ ontology and Chapter 6 introduces S-BOUN-TI, the approach for machine interpretable topic identification from microblog post sets. Chapter 7 presents the related work and compares BOUN-TI and S-BOUN-TI. Chapter 8 discusses the results obtained from both approaches. Chapter 9 presents the conclusions of this thesis.

2. LITERATURE REVIEW

Topic identification and automatic text summarization refer to the task of identifying what a document or set of documents is about. Information retrieval techniques widely utilize topic identification methods. Before the Web, topic identification methods were applied to plain text documents [43,44]. After the emergence of the Web, attention shifted to identifying the topics of the Web documents (web pages) [45]. Especially, retrieving the Web documents of interest led to the development of Internet search engines. These engines identify what the topics of a web page are and index pages accordingly for further retrieval queries.

After the emergence of the Web 2.0, where user generated content can be published on the Web easily, effective topic identification of web documents became more important. One of the most important developments allowing the general user to put out information on the Web is blogs. Blogs are essentially online diaries that anyone can read and comment on. Effective topic identification methods have helped improve the retrieval process for blogs [46].

Another development of Web 2.0 applications is microblogging platforms. On these platforms, people write short messages about what they are doing, what is happening around them, or anything they would like to communicate quickly [47]. Identifying the topics of microblogs is even more interesting than before because unlike blogs, which are often written by only one person and are usually long, microblog platforms provide short messages by many users quickly. Twitter is one of the most widely utilized microblogging platforms. After the introduction of Twitter, two tracks of research emerged. While the first track focused on identifying characteristics of microblogs such as why and how people post, the second track focused on what the users are talking about. In this chapter, the work in the second track will be examined in detail.

The approaches in the second track differ in several in terms of whether they focus on a single or multiple posts, the use of external resources (such as WordNet), the methodology, and how they consider the resulting topics. The common methods are: probabilistic topic model such as LDA, classification of posts according to several features, clustering a graph where posts are nodes and edges are weighted according to a similarity measure between the

two posts. Similarity measures and classification features are based on the number of common words and the semantic relatedness of words or phrases - computed by using external data such as WordNet or Wikipedia - among posts. Meta information about microblog posts is also used in similarity measures and classification features.

Some approaches rely on external resources or meta information to assist in determining topics, whereas others do not. Typical external resources provide information about synsets of words (e.g. WordNet), encyclopedic information (e.g. Wikipedia), or slang (e.g. Twitter Dictionary Guide [48]). Meta information is information such as the language, creation-time, location, etc. of the post, which can also be useful in determining topics. For instance, the frequent change of words close together in time may indicate an emerging topic.

Some approaches focus on determining the topic of a single microblog post, while others attempt to determine the topic of a set of posts, such as those within a given time interval or geographical region. These approaches are referred to as single post processing and collective processing respectively.

Another research track uses manually defined or automatically obtained word or phrase sets for identification. A set represents a topic of interest. The number of times particular words or phrases in that set appear in microblog posts measures the strength of the topic.

In 2001, the "Semantic Web" was introduced by Tim Berners-Lee *et al.* [49]. The aim of the Semantic Web is to enable services so that machines can interpret the data of other machines. After the introduction of this concept, Resource Description Framework (RDF) was introduced as a standard to express data in machine interpretable form. It allows for the expression of resources and the relationships between resources. The Sparql Protocol and RDF Query Language (SPARQL) was introduced to enable the querying of data in the Semantic Web. This era of semantically expressing resources and publishing them on the Web has been referred to as the Web 3.0 era [50].

While it is useful to represent data in a machine processable way, how the data should be interpreted is not sufficiently represented by the RDF. All data does not have the same interpretation rules; interpretation is often determined by the domain the data is utilized

in. The expression of domain specific knowledge and models was required, and to meet this need ontologies were introduced. The Web Ontology Language (OWL), which is based on RDF, was used to represent ontologies. While these ontologies were developed for modeling knowledge in various domains such as health and biology, encyclopedic data which was not limited by any particular domain was also modeled by DBpedia [51]. Often locating this model at the center [52] by connecting data it offers to data in other domains, the Linked Open Data (LOD) project [53] was introduced to connect open and distributed data across the world. Since the data was open and accessible in a structured form with the expression of ontologies, the identification of meaningful parts in texts and linking these parts to definitions in the Semantic Web became a task that many researchers were interested in. This linking operation, referred to as "entity linking", made the semantics of a text more refined. As a result of linking in documents, complex queries became possible, such as "return all documents mentioning politicians". The entity reference comes from the entity linking operation, and the occupation of the entity is defined in the knowledge base. Entity linking is also referred to as semantic tagging, semantic annotation, and semantic information extraction.

Entity linking has been applied [54–56] to mediums such as news documents, meeting reports, and blogs. Approaches have been proposed that define an ontology or use existing vocabulary or ontologies to represent the information they extract. Among these approaches, some of them [57, 58] automatically semantically link meaningful parts of news documents to entities using natural language processing (NLP) techniques and express the extracted information in the Semantic Web. Entity linking has been applied to biological literature by Müller*et al.* [59]. Their approach identifies indicating words in documents and maps them to classes of an ontology they have defined. Using the definitions in the ontology enables domain specific query-based document retrieval. Some other approaches semantically annotate semi-structured documents. For example, the approach of Aggelen *et al.* [60] semantically annotates meeting reports of The European Parliament. The annotations are linked to DBpedia, GeoNames, and Eurovoc thesaurus. It automatically links by seeking matching strings of the labels in DBpedia. However, the framework that the authors propose includes manual checking by humans, which makes the operation of identifying entities semi-automatic.

Another approach [61] extracts words, terms, and concepts from documents, and expresses them using OWL and RDFS structures. LOD, Wordnet, and DBpedia resources

are used to represent terms and concepts. This approach is based on the Latent Drichlet Allocation (LDA). It analyzes the output of LDA topics and the input documents, forms related terms and nouns from LDA topics and expresses them in the Semantic Web. These approaches work on semi-structured documents such as meeting reports and text documents such as news and blogs. Detailed information about LDA is provided following this chapter.

With the help of entity linking approaches, identifying meaningful parts of texts and linking them to existing defining resources has made the processing of documents easier because, instead of words and phrases, the software agent that deals with the document can directly refer to a concept in an unambiguous way. Furthermore, entity linking can be used in topic identification, as topics are based on related elements and the elements are based on the meaningful parts of documents. Entity linking may also be useful for identifying and unifying meaningful parts. For example, the terms "Obama" and "Barack Obama" refer to the same entity (person). If this is identified, the topics that are output would be more precise.

This thesis focuses on identifying topics of microblog posts, and entity linking in microblog posts may play an important role in identifying the elements of topics. However, approaches that work on conventional texts can be challenging to apply to short texts like tweets, as their limited context and special jargon makes the identification task challenging. The approaches [10–14] link the text of single microblog posts or parts of single microblog posts to external resources. The approach of Ferragina *et al.* [10, 11] considers Wikipedia titles and link structures, while the approach of Gattani *et al.* [12] considers other data sources such as MusicBrainz, City DB, Yahoo! Stocks, Chrome, and Adam in addition to Wikipedia titles. The approach of Meij *et al.* [13] considers Wikipedia article bodies, links, and anchors in addition to titles for topic identification in a single-microblog post. Section 3.5 gives more information about entity linking.

Approaches have been proposed by Kapanipathi *et al.* [62] and Sahito *et al.* [63] to semantically represent a single microblog post in the Semantic Web. These approaches produce an RDF representation, often expressing the meta information about the post such as the author and date of creation, and use entity linking to identify meaningful parts and represent them in the RDF.

2.1. Approaches that provide words or phrases as topics

The related elements that are used to identify topics are often words or phrases. In the Twitter domain, hashtags (Section 3.1 gives details) are also used to indicate topics. Thus, in addition to words and phrases, the hashtags are also provided. The elements of other approaches are representative posts.

One of the widely utilized methods is Latent Drichlet Allocation LDA [64], which is a probabilistic topic model. The elements of LDA topics are words. The input of LDA is a set of posts and the outputs are the topic model, which includes the topics and their words. LDA-based approaches have been proposed by several works [30–34, 65].

In probabilistic topic modeling, documents are assumed to be a multinomial mixture of hidden topics, while the topics are represented as a probability distribution over a number of words. Topics are created, words are assigned to topics and topics are assigned to documents. Figure 2.1 shows sample assignments of topics to documents and words to topics. Often, topics are assumed to be assigned to a few documents, and some words are assumed to be assigned to topics. LDA gets the number of topics (T), and the Drichlet distribution parameters α and β that are used in assigning words to topics and topics to documents. α states assumption of document similarities. While a value close to zero assumes that the documents are not similar, a value close to one assumes that the documents are similar. Similarly, β states assumption of topic similarities. While a value close to zero assumes that the topics are not similar, a value close to one assumes that the topics are similar. The common practice is setting low β values such as 0.01 and 0.001. The α value and the number of topics (T) are often domain dependent and tuned through experience.

LDA starts by assigning words to topics randomly. Then, it updates the assignments iteratively. In each iteration, for each word assignment to a topic, and topic assignment to a document, new assignments are sought, and the iterations are expected to converge.

The assignments are made according to the following: Let $n_{d,k}$ be the number of times document d refers to topic k. Let $v_{k,i}$ be the number of times topic k is assigned to the word w. Let $v_{k,w_{d,n}}$ be the number of times topic k is assigned to the word w because of

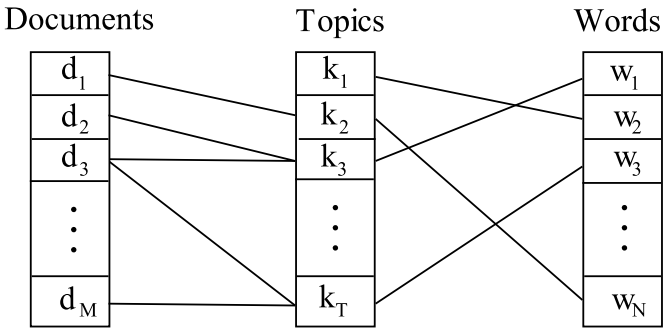


Figure 2.1. Sample assignments of documents, LDA topics and words.

the occurrences of w in document d. Let α_i be the Drichlet parameter for the distribution of topic i. Let β_i be the Drichlet parameter for the word distribution of word i. The likelihood of assignment of word w to topic k is computed as in Equation 2.1.

$$\frac{n_{d,k} + \alpha_k}{\sum_i^K n_{d,i} + \alpha_i} \frac{v_{k,w_{d,n}} + \beta_{w_{d,n}}}{\sum_i v_{k,i} + \beta_i} \tag{2.1}$$

Applying LDA to microblog posts is challenging if each post is considered as one document due to their brevity. To address this issue, several approaches have been proposed. One type of approach involves concatenating short texts to obtain larger documents. This type of application often assumes that the posts are related in a context, such as posted close together in time and/or location [66], posted by the same user [67], or having the same special context indicator (hashtag) [65]. Another type of solution to the short text problem, which enriches documents with co-occurrence information, is given by Yan *et al.* [33]. The authors considered documents as a composition of co-occurrences of word pairs. This approach therefore considers topic elements as word pairs. For example, their approach considers a four word document as if it has six words (six pairs of words).

The topics of the LDA can be assumed as the set of words (the words are assigned to topics). For instance, Zhao *et al.* [34] gives the set "rob, moon, love, twilight, gaga, lady, #nowplaying, adam, lambert, fans, kris, chris, brown, song, beyonce, download, live, mixtape, music" as a topic. To refer this topic to a concept, the authors further process and automatically assign it to "arts" from a set of pre-defined categories.

Applying LDA to microblog posts brings several limitations. LDA requires parameters α, β, and T (the number of topics) to be set, which often requires manual observation.

Topics of LDA do not fully describe microblog post topics. Identifying the underlying concept associated with sets of words usually requires additional - possibly manual - analysis.

Another approach by Yan *et al.* [68], similar to LDA, tries to find the latent topics in data represented by the document-term frequency matrix. This approach is based on the factorization method non-negative matrix factorization by Lee and Seung [69]. For the non-negative matrix factorization task, if the whole is represented as a matrix, and if it is assumed that there are two parts, the parts are assumed to be represented by two matrices whose product makes the matrix that represents the whole. If this is applied to topic identification, the document-term frequency matrix is factorized into two matrices which represent document-topic and topic-term matrices. The number of topics is an input for this operation. Matrix factorization is often approached as an optimization task that at the end produces two matrices whose product is a matrix which is similar to the original matrix.

Another type of approach that provides related elements considers the highly temporal nature of microblog posts. The approaches of Alvaniki *et al.* [16], Cataldi*et al.* [17], kasiviswanathan*et al.* [19], Trilling *et al.* [22] and Chen *et al.* [18] base topic identification on the frequency of change in words and hashtags. The words become topic elements and are presented as topics.

Among these approaches, the approach of Alvaniki *et al.* [16] computes the relatedness of two elements (hashtags or words), and monitors this relatedness over time. A deviation of the relatedness value is considered as an emerging topic. The deviation is computed on a sliding window. The relatedness of two elements is computed with Jaccard similarity ($Jac(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2}$) and the global importance in the current window ($Imp(S_1, S_2) = \frac{S_1 \cap S_2}{n}$) where S_1 is the set of documents (posts) that has the element e_1, S_2 is the set of documents (posts) that has the element s_2, and n is the number of documents.

The likelihood of the next value of the relatedness values is predicted, and the difference between likelihood of the next value and the actual value determines whether the two elements belong to a topic. The likelihood of the next value is computed as $v'_t = \alpha v_{t-1} + (1 - \alpha)v'_{t-1}$ where v'_t is the prediction score, v_{t-1} is the latest observed value, and v'_{t-1} is the latest computed likelihood. The α value is set as favoring the recent observation. The approach

considers two elements as one topic if the difference of the latest observation and the latest prediction is above a threshold. If three elements (e_1, e_2, and e_3) are considered, and if e_1 and e_2 belong to the same topic, e_2 and e_3 belong to the same topic, and e_1 and e_3 co-occur in the same post at least one time, then the two topics are merged which results in e_1, e_2 and e_3 being put into one topic.

Other approaches that output a set of words, phrases or hashtags are based on a similar idea of comparing new content with recent-past content. For example, the approach of Kasiviswanathan *et al.* [19] uses dictionary learning, which is a machine learning approach, to model recent content. Similarly, the approaches of Cataldi *et al.* [17] and Chen *et al.* [18] use the time and co-occurrence of elements in the same microblog post to identify topics. These approaches also take into account the authority of the users who post using page rank algorithm.

The approach of Lehmann *et al.* [70] identifies potential topics using WordNet [71]. This approach takes a set of posts and outputs topics. The elements of the topics of this approach are words, and the matching rate of words in posts are computed with the words in Wordnet synsets. The topics are referred to as the synset classes of words. The authors manually investigated the results and discovered that temporal patterns of hashtags indicate different Wordnet classes.

2.2. Approaches that manually define topics

Another collective processing approach is outlined in the study of Lansdall-Welfare *et al.* [6]. The input of this approach is a set of microblog posts and the elements of topics are the words. This approach assumes that the keywords of topics are manually specified. Counting how many of these keywords appear in the post set during a specific time interval determines the strength of the topics. In application, the authors identify keywords for four classes of moods. Among the defined mood classes, the anger class has 146 words, the fear class has 92 words, the joy class has 224 words, and the sadness class has 115 words. Posts are processed with the assumption that if a keyword is found in a post, it was written by a user whose mood the keyword is associated with. For each mood, the results are aggregated and public moods over a time period are shown. In the study, Lansdall-Welfare *et al.* were able

to identify Christmas, Halloween, Valentines Day, and Easter along with attributed moods for those days. They were also able to identify overall negative public moods such as those that occurred after cuts in public spending. A similar approach has been applied by Prieto *et al.* [38]. This approach specified the words and regular expressions of topics manually. The input of the approach is a set of microblog posts, and each group of words and regular expressions indicate a sickness. The importance of a sickness to the public is measured by counting the matching words and regular expressions in a post set. When the posts related to sickness are identified using the manually defined words and regular expressions, they are clustered into four categories such as depression, pregnancy, flu and eating disorders. From these categories, the authors manually selected distinguishing features for machine learning training. They could effectively classify posts into groups, such as if they were flu or eating disorder related or not.

In contrast to these approaches, indicative words in the domain of health were automatically obtained in the approach of Parker *et al.* [39]. The input of this approach is a set of posts. The topic elements are words, and they are represented by Wikipedia pages. The authors extracted words (e.g. symptoms) from health-related Wikipedia articles. These words are considered to be indicative words of topics, and the topics represent public health trends. Counting the words related to a particular topic (sickness) in a post set measured the importance of that sickness to the public. Using indicative words, the posts are processed, and posts containing one of these words is assigned to its sickness class. The authors applied the approach on health related post sets and showed that they could identify sicknesses like influenza, summertime ice cream headaches and allergies.

2.3. Approaches that provide representative posts as topics

Alternatively, regarding the generation of possible topics, other approaches present topic elements as posts. The approaches of Mathioudakis and Koudas [21] and Marcus *et al.* [20] keep track of frequencies of recent words in posts. If the frequency of a word increases a lot (such as from about 5 posts per minute to about 100 posts per minute) then the element is considered as indicating a possible topic. To form topics, co-occurrence in the same microblog post is considered. If two words have recently co-occurred in the same post several times, they are grouped together. After identifying the words, representative posts are selected to be

shown to the user along with the words. The selection is based on the number of occurrences of the words in posts [21] and number of matches in the group of words, where the group is based on co-occurrence in the same post [20].

2.4. Approaches that provide summarization phrases as topics

In addition to topic identification approaches that provide topic elements as words, phrases, and representative posts, another research track provides semi-readable phrases that indicate what people are talking about [1, 72]. A well known phrase construction method in microblogs is proposed by Sharifi *et al.* [1]. The input of the approach is a microblog post set and a phrase of interest. The approach seeks overlapping phrases. It starts by generating a graph with one node that represents the phrase of interest. It adds nodes to the graph representing the words to the left and right of this node. The Figure 2.2 gives an example graph.

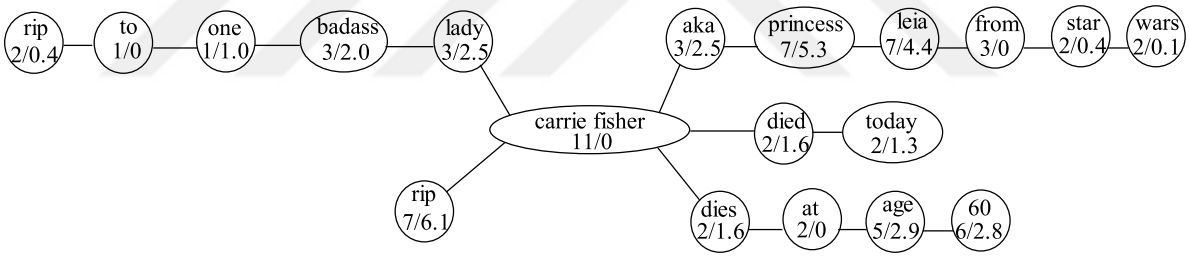


Figure 2.2. An example graph for summarization that can be obtained by the approach of Sharifi *et al.* [1]

In this graph, the nodes have two values. The one on the left is the frequency ($tf(n)$) of the word of the node n in that sequence of nodes. The value on the right is the weight of the node. The weight of the node n is computed as $w(n) = tf(n) - dist(n)log_b\ tf(n)$ where $dist(n)$ is the distance between the node n and the node of the phrase of interest at the center. For the stopwords, and the words that appear in every post (such as the phrase at the center) the weight is set to zero. The value of b determines the length of the summaries generated by the approach. While lower values such as 2 and e lead to smaller lengths of summarization, higher values lead to longer lengths.

Once the graph is constructed, the approach is to search for the most overlapping phrase. Using a depth first search on the right and the left subgraphs, beginning from the

center node, the path with the most weights is searched and output as the summary of the post set. For this example, the most weights are obtained from the phrase "rip carrie fisher aka princess leia from star wars". The phrase is reformatted to resolve the issue of improper cases such as the usage of "rip" which is often in upper case: "RIP". For this, the posts are searched and the post substring that best fits the phrase is chosen. For this example, "RIP Carrie Fisher aka Princess Leia from Star Wars." is the final output.

The authors inspected the results and realized that there are cases that need improvement. To improve the algorithm they updated the weighting of words. The weight of n is $w(n) = tf(w)idf(w)$ where $idf(w)$ is the inverse document frequency which is computed as $idf(w) = log_2 \frac{number\ of\ all\ posts}{number\ of\ posts\ that\ has\ w}$, and $tf(w)$ is the term frequency of w in all posts. They showed that weighting nodes with this metric resulted in better summarizations. With this measure, the authors employed a different *tf-idf* computation. Normally for computing *tf* values, the frequency of the term is computed from one document. In their computation, the frequency of the words are obtained from all documents, which are all microblog posts in the set.

This approach builds the summarizing phrase towards the left and right considering the phrase of interest placed at the center, using common consecutive words. However, since the phrase is formed from microblog posts, sometimes it may not be grammatically correct or even meaningful.

2.5. Approaches that classify posts to extract domain specific information

And finally, the last research track is the identification of domain specific information such as "the mood of the user" or "reporting of earthquake occurrence" using machine learning techniques. One of these studies by Schulz *et al.* [23] classifies a post according to whether it is incident related or not. Incident related tweets are grouped under three different categories, which are "crash", "fire", and "shooting". The authors showed that they were able to identify whether a post reports an incident. Another approach, by Sakaki *et al.* [9], classifies a post if it reports that an earthquake or typhoon is occurring during the time it was posted. In this approach, posts are considered as sensors that indicate if an earthquake or typhoon is happening. First, the authors analyzed posts and found useful ones that report an earth-

quake or typhoon is happening. Then, features are extracted from these posts and a model is trained. They discovered that, for earthquakes, shorter posts such as "Earthquake!!", or "its shaking" are more effective in identification.

Another group of domain specific information extraction methods are geared to the task of sentiment analysis. For sentiment analysis, a sentence or a phrase is often classified into "positive mood", "negative mood" or "neutral". In the microblogging environment, sentiment analysis is often applied to single posts, or to terms or phrases in posts. Several approaches have been used for microblog posts [15,40,41]. The features used for classification are usually words (uni-grams), (bi-grams), abbreviations and emoticons. In this domain, emoticons and abbreviations are often considered as indicative features of sentiments.

The approaches in this research track can be applied to post sets. For a given set of posts, the approach can be applied to each post, and the results can be aggregated to obtain results such as public health trends [39], public mood changes [6], earthquake time and location detection [9].

2.6. Conclusions

In conclusion, the literature related to identifying the topics of microblog users' posts consists of several types of work. While some of the approaches focus on single posts, other approaches focus on identification using a set of posts. While some approaches use external resources such as Wikipedia for identification, others do not. The type of output of the different approaches also varies. While some output a set of words or phrases as topics, others output a set of representative microblog posts. One type of approach that uses external resources is entity linking, which is a single post approach.

Once the literature is reviewed, it can be observed that there is a lack of approaches that work on multiple posts but also identify using external sources such as Wikipedia and DBpedia. It can also be observed that the output of these approaches are neither human readable nor machine processable; they require further processing by both humans and machines in utilizing the outputs. BOUN-TI, which is introduced in this thesis, fulfills the human readable topic identification need. It also works on multiple posts, utilizing an external re-

source. S-Boun-TI, which is also introduced in this thesis, fulfills the machine interpretable topic identification need. It identifies topics of multiple posts, expressing the topics in the Semantic Web, which makes the topics machine interpretable.

3. BACKGROUND

This section provides foundational information about several methods, tools, resources, and services key to understanding this thesis. In the context of this thesis, the terms "tweet", "microblog post" and "short message" are used interchangeably, because the approaches introduced in this thesis can be applied to other short message systems.

3.1. About microblog posts

Microblog users tend to write short, expressive, and distinctive messages. Especially if users want to be noticed, they need to make effective use of the limited space provided. For instance, users employ hashtags, intending to make their posts share the same context with posts using the same hashtag (hashtags start with # sign). Consider the tweet "Obama: Take some of the money we're saving as we wind down two wars to rebuild America. #debate". This tweet declares that its subject is part of the context of the debate, and uses one of Obama's quotes. In this way, the author makes a connection between Obama's words and the context of the debate. Further, he can add his opinion on the subject if he wants.

Users intend to post frequently about various topics, and this makes microblogs a useful source for investigating the intentions of a particular group of people. Regardless of potential usefulness, however, it may be difficult to process tweets due to different posting styles. Tweets are full of abbreviations (hny for honey, omg for oh my god), misspellings, jargon, profanity, and Twitter specific syntax (i.e. hashtags like #cantwait and user references @camanpour). They are often one or more fragments of sentences. These factors make it difficult to process posts with conventional Natural Language Processing (NLP) approaches [4].

3.2. Twitter API

Twitter provides an API [73] for retrieving tweets in a computer program. It returns tweets for given specific term(s) or phrase(s). It also provides a public stream, which gives a sample from all tweets. Twitter applies rate limits to the amount of tweets that can be retrieved for a specific amount of time. A tweet is returned as an object which can be

requested in Javascript Object Notation (JSON) format. In addition to 140-characters text, a tweet object has other properties such as information about the author of the tweet, the posting date, time, shared links in the tweet text and much more. Tweet posters use special syntax to refer to other Twitter users using @ signs, and tag with hashtags using # signs to refer to the context of tweets that use the same hashtag. Twitter API also returns hashtags and user mentions as properties of tweet objects.

3.3. Wikipedia

Wikipedia includes over five million articles about many topics. Articles are formed by human contribution and collaboration. Wikipedia policy dictates that the title of an article is either a name or a description of the subject of the article [74]. Therefore, the title of a Wikipedia article can be used to refer to similar content.

3.4. Vector space model and document similarity

The *tf-idf* vector space model represents a document (in a set of documents) as a vector [75]. Each element in the vector has a value of a word that gives the strength of the relation between the word and the represented document.

The strength of a word in a document is obtained by the product of the term frequency (tf) and the inverse document frequency (idf). tf is the number of times the word occurs in that document and idf is obtained as $idf = log\left(\frac{number\ of\ all\ documents}{number\ of\ documents\ which\ include\ the\ word}\right)$. The similarity between two documents is the cosine of the angle between the two representative vectors. The cosine similarity between the two vectors A and B is computed as given in Equation 3.1

$$cosineSimilarity(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i}\sqrt{\sum_{i=1}^{n} B_i}} \tag{3.1}$$

3.5. Entity Linking

With such wide usage of microblog posts, identification of real world entities in posts is needed. "Entity Linking" is the process of linking text fragments in documents to their real-world entities [55]. In the microblog domain, these documents are microblog posts. Entity linking in microblog post texts presents challenges due to the special syntax and evolving jargon as well as the limited context. Microblog users shorten posts or divide their longer expressions into several posts due to limited space.

An entity linker extracts spots (surface forms) in the text. Spots are meaningful fragments of texts which are candidates for linking. Spots are often identified using part of speech (POS) tagging methods that help identify nouns which are candidate spots. An entity linker chooses entities from a set of entities, often using syntactic matching or statistical methods such as choosing the most popular ones. From the reduced set of entities, it picks the most suitable entity. In this step, statistical methods or more sophisticated machine learning methods are often applied. The shortness of posts limits the context used to disambiguate an entity link among several candidates.

TagMe [11] is a widely utilized state of the art tool, designed for annotating fragments of short texts. It returns two confidence values which are the confidence of text fragment being a spot (link probability p), and the confidence of the linking of the entity ρ. TagMe provides a RESTful API [76]. Figure 3.1 shows an example of the spots and links provided by TagMe for a tweet text. The ρ values are shown above each spot, which indicate the confidence that they are spots. Entities from Wikipedia, such as "https://en.wikipedia.org/wiki/United_States_Department_of_Justice" are presented along with the probability (p) that it is a good link.

In this thesis, entity linking is referred to as the annotation of spots with Web resources.

3.6. Linked data

Linked data [53] is concerned with making related available Semantic Web resources linked. A linked data crawling study in 2014 [77] reports that they crawled 900,129 documents

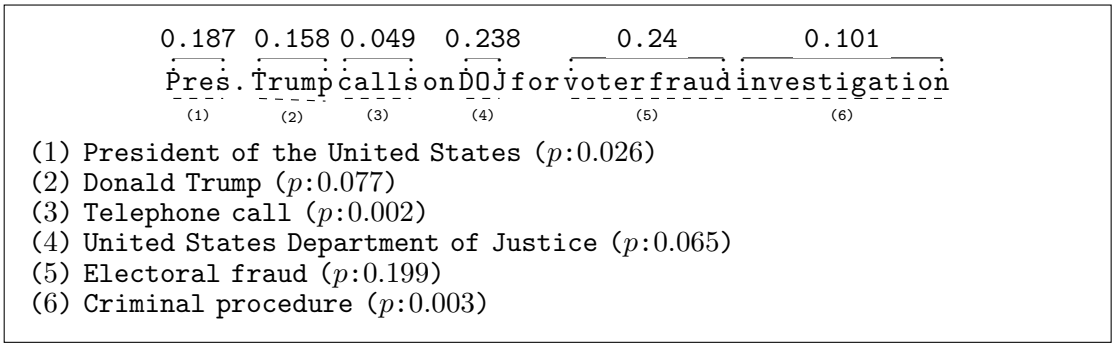


Figure 3.1. Six spots and their associated entities as suggested by TagMe for a given tweet text.

describing 8,038,396 resources. The resource descriptions mostly use Resource Description Framework (RDF), RDF Schema (RDFS), Web Ontology Language (OWL), Friend of a Friend (FOAF), and Dublin Core Metadata Initiative (DCMI), Semantically Interconnected Online Communities (SIOC) and Simple Knowledge Organization System (SKOS) terms and statements.

One of the data sources in Linked Data is DBpedia [51]. DBpedia has encyclopedic information derived from Wikipedia. DBpedia is enriched with Yago classes [78], schema.org vocabulary [79], Geonames and FOAF [80] ontologies. Another common knowledge platform is Wikidata [81] with over 24 million resources. Wikidata is a collaborative knowledge creation and editing platform. Both Wikidata and DBpedia have SPARQL endpoints [82, 83] which make them queryable online.

3.7. W3C Vocabularies and Ontologies

FOAF [80] is a vocabulary which is commonly used to express agents, with an emphasis on people and relationships among people. DCMI terms (dcterms) vocabulary is often used to describe meta information about a resource such as title, creator, description, date etc., and schema.org vocabulary is a vocabulary that defines common structures such as events, health, organizations, persons, products, and businesses.

Geo-location vocabulary [84] is a W3C standard which is used to express the longitude and latitude of spatial things. Geonames [85] is a more detailed ontology that defines geolocations. In addition to longitude and latitude, Geonames provides the relation of one

location to another, such as being nearby or a part of another location. Each geolocation is an instance of "Feature" class. Geonames also provide a database of over 11 million instances of the "Feature" class all over the world.

W3C OWL-Time ontology [86] provides expressions of temporal concepts about entities. Currently, the latest version of the ontology is published, although the latest version of the introducing document [87] is in the working draft stage. The documentation states that the ontology development is backwards compatible.

3.8. Ontology, vocabulary and data namespace prefixes

This thesis refers to several ontologies, vocabularies and data name spaces. Each of them are listed in Table 3.1. The prefixes are formatted in "prefix:" format which is widely used in Semantic Web documents and queries such as SPARQL. Each prefix is listed with the corresponding Internationalized Resource Identifier (IRI) (RFC3987 [88]) which is a generalization of Uniform Resource Identifier (URI) (RFC3986 [89]). A sample prefix list is given by W3C in the SPARQL documentation [90]. This thesis uses prefixes to shorten URLs. For example, "dbr:Hillary_Clinton" is the same as the URL "http://dbpedia.org/resource/Hillary_Clinton". In this example, "dbr:" corresponds to "http://dbpedia.org/resource/".

3.9. Tools

Four tools were employed for the work outlined in this thesis. Three of them, PostgreSQL, Solr and Fuseki, are used to index various types of data. Phirehose is used to retrieve tweets from Twitter.

PostgreSQL [91] is a database management system which is widely used to store relational data.

Solr [92] is a document indexing tool based on the Lucene [93] index. It provides a web service that serves documents and makes them accessible trough queries.

Table 3.1. The ontologies and data name spaces referred to in this thesis.

Prefix	**IRI**
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
xsd:	http://www.w3.org/2001/XMLSchema#
geo:	http://www.w3.org/2003/01/geo/wgs84_pos#
geonames:	http://geonames.org/ongology#
foaf:	http://xmlns.com/foaf/0.1/
time:	https://www.w3.org/2006/time\#
schema:	http://schema.org/
dbr:	http://dbpedia.org/resource/
dbo:	http://dbpedia.org/ontology/
dbp:	http://dbpedia.org/property/
umbel:	http://umbel.org/umbel#
dbpedia-wikidata	http://wikidata.dbpedia.org/resource/
dcterms:	http://purl.org/dc/terms/
topico:	http://soslab.cmpe.boun.edu.tr/ontologies/topico.owl#

Fuseki [94] is an open source web application that serves RDF data. It has a built in OWL reasoner. It allows querying RDF data using SPARQL.

Phirehose [95] is a PHP library that is used to retrieve tweets from Twitter. The library provides access to the Twitter Streaming API (firehose, etc).

4. TOPIC IDENTIFICATION USING WIKIPEDIA

In a set of microblog posts, there are likely numerous topics being discussed. The challenge is figuring out what they are. External resources such as Wikipedia provide a broad spectrum of topics which can be used to represent microblog topics. While microblogs are a suitable resource for user generated content about topics and issues people are interested in, Wikipedia also contains user generated content on topics of interest to many people. Wikipedia also has a wide spectrum of articles, including recent ones [12, 14]. Most of the earlier work that refers to topics using external sources such as Wikipedia has focused on extracting the topics of single tweets. The challenge of this study is dealing with collections of tweets.

This chapter describes an approach, BOUN-TI, that attempts to identify topics of sets of microblog posts by examining the posts and mapping them to Wikipedia article titles. One of the main questions is whether elements in the texts of microblog posts can be related to Wikipedia articles. The former are messy, noisy and unstructured, whereas the latter use well-written formal encyclopedic language. It would be interesting to examine whether a relation can be found between these different mediums.

Modeling topics as Wikipedia article titles has the advantage of being human readable, unlike models that present keyword sets that require further processing. Also, there is no preset number of topics as input in this method.

The proposed approach extracts topics from post collections representing distributed pieces of information in the post collection. While a single post may be about several topics, it may also include valuable aspects of a topic related to the post collection. Considering that posts are short, these aspects are distributed over multiple posts. The term “topic”, here, is used to mean the concepts underlying the given microblog post set.

The issues that need to be resolved are related to how to select relevant articles for a given set of posts, with the idea being to locate the articles that are closest to the content of posts in the given set. Whether using a simple bag-of-words method and cosine similarity

based comparison is suitable is being investigated.

The comparison of the content between a Wikipedia article and a post set shows the scattered distribution of pieces of information in microblogs. For example, a post set may consist of many posts having the word "abortion", and many other posts having the phrase "Catholic Church" with a smaller number of posts having both of them. If this set is processed as a collection, but not individually, the Wikipedia article title "Christianity and Abortion" would be a result. Short document or single post processing approaches [10–13] that use external sources would result in the separated topics "Catholic Church" and "Abortion" in this case. The aim of these approaches is to enrich short text content with semantics by linking meaningful parts of the text to external sources such as Wikipedia or DBpedia. Aggregation of the results of these approaches such as computing frequencies of linking of the external sources in the post set would result in a ranked list of external sources. However, this approach would miss the relation among entities, which would lead to missing external sources that may represent these relationships (such as "Christianity and Abortion" rather than "Catholic Church" and "Abortion" given as topics). Therefore, by examining a collection of posts as a whole, it is possible to identify terms related to a topic that pertains to the whole set. Furthermore, the expectation is that the significant terms will also appear in Wikipedia articles – the encyclopedia by the people, for the people.

This work was reported by Yıldırım *et al.* [96]. The remainder of this chapter is as follows: Firstly, the approach is detailed in Section 4.1. A prototype is implemented to evaluate the approach. The implementation details of the prototype are given in Section 4.2. Experiments and results are presented in Section 4.3. Observations about the topics that are identified by the approach are given in Section 4.4. Lastly, conclusions and future directions related to this approach are given in Section 4.6.

4.1. Boun-TI: Topic Identification Using Wikipedia

Considering that Wikipedia articles are candidate topics, human readable topic identification can be considered as an information retrieval task that is defined as "return best representing topics of a microblog post set among candidate topics". The post collections may be retrieved in various ways, such as using the search and public stream APIs [73].

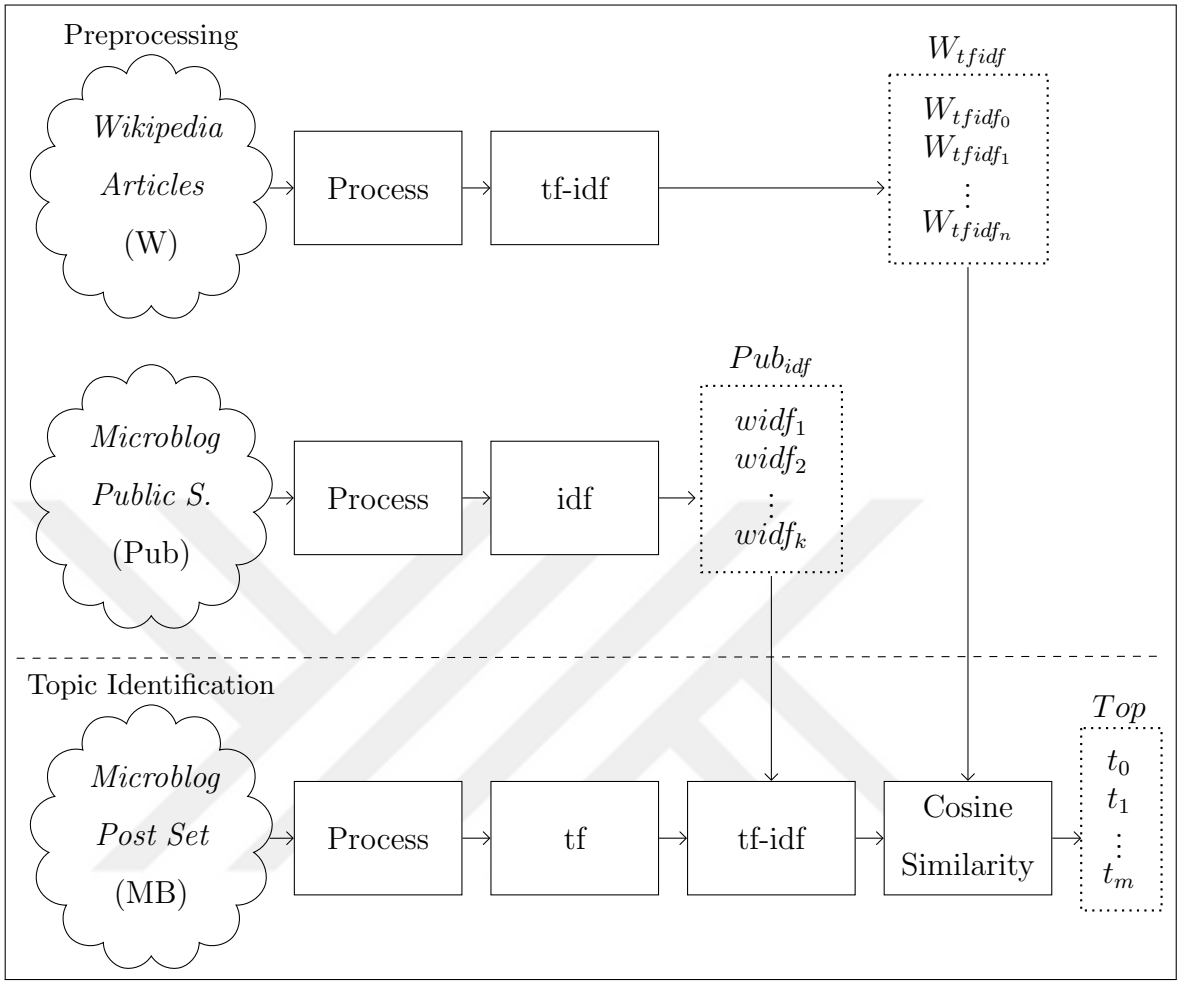


Figure 4.1. Overview of the process of topic extraction from a set of microblog posts.

The proposed approach's overview is given in Figure 4.1. The approach has two main parts. The values that will be used in identification of topics by checking similarities are stored in the preprocessing part. The topic identification part does the computation on the input microblog post set by using the resources created in the preprocessing part and extracts tokens. The "Process" boxes represent several operations used to filter out content that is not needed during the computation. Links are removed. For hashtags, camel-cases are resolved. Abbreviations, special syntax, and profanity that are used in microblog posts are not needed in this context since Wikipedia articles are more clear. Only alphabetic characters are considered as tokens.

The basic approach is to apply *tf-idf* vector space model to microblog post sets and Wikipedia articles. Cosine similarity is performed to the vector of the microblog post set and a Wikipedia article. This tells us how similar the Wikipedia article and the post set are.

In conventional *tf-idf* vector space computation, all documents are represented as a vector of the size of all words in the corpus. Directly applying this approach to solve this problem suggests that the *tf-idf* values of representative vectors of all Wikipedia articles should be re-computed. With each post set arrival, the corpus of all documents changes because the new arriving post set may come with new words. Likewise, new words may change the vector sizes of articles. The *idf* values of lots of words (those in the microblog post set) would slightly change, and this change would effect Wikipedia articles' vector values. This computation would take too much time due to the huge number of Wikipedia articles (at the time this study was conducted there were over 4M articles, with over 5M at the time of publishing). While Wikipedia itself is getting updates, although not as frequently as microblogs, it is accepted as unchanging, as re-computing all values would be required.

In order to overcome the problem of re-computation of *tf-idf* values of the words of Wikipedia articles, the values are computed once and stored. When a comparison is needed between a Wikipedia article and a post set, their words and *tf-idf* values are put into a vector by forming it on the fly. The size of the created vectors is the size of the union of the words of the Wikipedia article and the post set. Each element of the vectors with the same index represents a unique word in the union set. For each element of the vectors, corresponding *tf-idf* values are assigned. If one side does not have a word, its value is set to zero. This operation allows the two documents to be represented in the same vector space.

While applying these operations, the values are kept as a set instead of a vector. The name of a set is given to indicate whether it is a *tf* set, *idf* set or *tf-idf* set, such as Pub_{idf}, MB_{tf}, and W_{tfidf}. The set W_{tfidf} is one of these sets. It has elements of sets of (word,*tf-idf*) pairs, where each set corresponds to a Wikipedia article.

In order to measure the similarity of a set of microblog posts to a Wikipedia article, the tokens of posts and *tf-idf* values should also be computed. Since microblogging is a different environment than Wikipedia, the *idf* values related to this environment are specific to it. Inverse document frequency (*idf*) is a measure that gives high scores if the word appears in a smaller number of documents and gives low scores otherwise; it is a measure of uncommonness. Thus, computing *idf* values of microblog words means computing how uncommon the words are in microblogs. The input post set is not appropriate for *idf* computation, especially if

the post set used for *tf* computation is retrieved using one or more keywords. In this case, the posts' content is dominated by the keywords, which makes keywords common for that post set. Since the *idf* is a measure of uncommonness, in this case these words will get lower values. This is not desirable because the topics these keywords are related to are of interest (which is the same reasoning behind using keywords in the first place). Decreasing the *idf* value for these words will result in lower scores for the related topics.

For this reason, an approach is needed to compute the *idf* of microblog words. This can be done by using a post set that is not dominated by a specific keyword, and it should be done for a sufficient amount of time so that the values of words are not affected by the coverage of the time interval. Microblog public stream in Figure 4.1 shows a post set. In the implementation, as with this set, BOUN-TI uses a large corpus which has a duration of five days of Twitter Public stream API sample endpoint [97]. Each public stream post is considered as a document for computing the *idf* of words. The value of each word is stored, and the set Pub_{idf} represents this store. Each element idf_i is a (word,*idf*) value pair.

All microblog posts in the input set are considered a single document when computing the *tf* values for the words. Thus, *tf* represents the number of times a word occurs in the set of posts. This is similar to the summarization approach in [1].

It is possible that some words may be extracted from the input post set, but cannot be found in the general public post set Pub. This happens when a word has not been previously encountered, and, thus, is uncommon. Its *idf* value is assigned as the maximum value, which is obtained if the number of documents for the word is set to 1. Therefore, the word is assumed to be previously encountered only once. The *idf* value of a word w in the public stream set (Pub) is:

$$\text{idf}(w, Pub) = \log\left(\frac{|Pub|}{max(|\{p \in Pub : w \in p\}|, 1)}\right) \tag{4.1}$$

Cosine similarity is a vector operation. In these operations, the documents (i.e.Wikipedia articles, and the input microblog post set) are represented as sets of pairs where each token is assigned a value. The cosine similarity computation is modified as a set operation to indicate the operations in the implementation.

The formal expressions of the computations are given below:

Let $\mathbb{N}$ be the set of natural numbers, $\mathbb{R}_{\geq 0}$ be the non-negative real numbers. A string is a sequence of alphabetic characters. Let S be a set of all strings.

Let set Pub_{idf} represent the set of tokens (strings) and their corresponding *idf* value pairs which is the Pub_{idf} in Figure 4.1. Pub_{idf} is a set of pairs (w, r) where $w \in S$, $r \in \mathbb{R}_{\geq 0}$, $Pub_{idf} \subset S \times \mathbb{R}_{\geq 0}$, and Pub_{idf} meets the constraint $\neg\exists(w, r_1), (w, r_2)\ [r_1 \neq r_2 \wedge (w, r_1) \in Pub_{idf} \wedge (w, r_2) \in Pub_{idf}]$.

Each value of $(w, v) \in Pub_{idf}$ is computed as shown below where Pub is the public stream posts.

$$Pub_{idf} = \{(w, v) | w \in S \wedge v = \mathrm{idf}(w, Pub)\} \tag{4.2}$$

Let set MB_{tf} represent the set of tokens (strings, or words) and their corresponding frequencies (*tf* values) in the input microblog post set. MB_{tf} is a set of pairs (w, n) where $w \in S$, $n \in \mathbb{N}$, $MB_{tf} \subset S \times \mathbb{N}$ and MB_{tf} meets the constraint $\neg\exists(w, n_1), (w, n_2)\ [n_1 \neq n_2 \wedge (w, n_1) \in MB_{tf} \wedge (w, n_2) \in MB_{tf}]$.

Let set C represent all possible tokens and their possible corresponding *tf-idf* values. Let set W_{tfidf} represent words and their *tf-idf* values for all Wikipedia articles, which is W_{tfidf} in Figure 4.1. C is a set of pairs (w, r) where $w \in S$, $r \in \mathbb{R}_{\geq 0}$, $C \subset S \times \mathbb{R}_{\geq 0}$. W_{tfidf} is power set of C. $W_{tfidf} = \mathcal{P}(C)$. And W_{tfidf} meets the constraint $\forall d \in W_{tfidf} \neg\exists(w, r_1), (w, r_2)\ [r_1 \neq r_2 \wedge (w, r_1) \in d \wedge (w, r_2) \in d]$.

Given sets Pub_{idf}, MB_{tf}, and W_{tfidf}, set $Top = topics(Pub_{idf}, MB_{tf}, W_{tfidf})$ is computed. It is shown as Top in Figure 4.1. Set Top represents Wikipedia articles and their corresponding similarity values to the input microblog post set.

$$topics(Pub_{idf}, MB_{tf}, W_{tfidf}) =$$
$$\{(d, s) | d \in W_{tfidf} \wedge s \in \mathbb{R}_{\geq 0} \wedge s = \mathrm{CosSim}(\beta(\alpha(Pub_{idf}, MB_{tf}), d), \beta(d, \alpha(Pub_{idf}, MB_{tf})))\} \tag{4.3}$$

$$\alpha(Pub_{idf}, MB_{tf}) = \{(w,r)|\exists v,n[(w,v) \in Pub_{idf} \wedge (w,n) \in MB_{tf} \wedge r = vn]\} \tag{4.4}$$

$$\beta(X,Y) = X \cup \{(w,0)|\exists p[(w,p) \in Y] \wedge \forall p[(w,p) \notin X]\} \tag{4.5}$$

$$\text{CosSim}(X,Y) = \frac{\sum\limits_{(w_1,r)\in X,(w_2,q)\in Y} \delta_{w_1,w_2} rq}{\sqrt{\sum\limits_{(w,p)\in X} p^2}\sqrt{\sum\limits_{(w,p)\in Y} p^2}} \tag{4.6}$$

$$\delta_{a,b} = \begin{cases} 1, & a = b, \\ 0, & otherwise \end{cases} \tag{4.7}$$

Equations 4.3 to 4.5 describe the cosine similarity operations and how the sets that represent cosine similarity vectors are constructed. The CosSim function (Equation 4.6) computes the cosine similarity given two sets' tokens and *tf-idf* value pairs. In the context of this computation, these sets correspond to representation of the input post set and a Wikipedia article. One of these sets is obtained from Twitter and the other from a Wikipedia article. The β function (Equation 4.5) makes the parameters for the CosSim function. It adds the tokens that exist only in the first set to the second set and assigns their *tf-idf* values in the second set to zero. The β function is called twice with the *tf-idf* sets that represent the tweet set and the Wikipedia article by swapping their order between the calls to add to both sets the tokens they lack but the other has.

The effect of adding tokens with zero values to the set that does not have that token, but which the set on the other side of the comparison has may seem unclear. However, this operation does not effect the results. The result would be the same if both sets were represented as vectors, because in the vector representation of tokens, the documents are represented by a vector of the size of the number of all tokens, where each element of the vector corresponds to the *tf-idf* value of a token. If any token is not included in a document, its *tf* is set to zero. This results in vector values of zero that each side of the comparison has. Zero values will not effect the computation since these tokens will add zero to the values summed in the nominator and the denominator of the cosine similarity calculation. Therefore, the cosine similarity function takes into account both if a token appears and if a token does not appear in the representative set. The common terms in sets contribute to increasing the similarity between the two sets. If a term does not exist in one of the representative

sets, it decreases the cosine similarity due to the normalization factor in the denominator of Equation 4.6.

In the following, an example is given in a simple environment. Let set Pub_{idf} consist of only the words "church, catholic, abortion, and health", and their corresponding *idf* values which is $Pub_{idf} = \{(\text{“}church\text{”}, 0.23), (\text{“}catholic\text{”}, 0.27), (\text{“}abortion\text{”}, 0.475), (\text{“}health\text{”}, 0.53)\}$. Let set MB_{tf}, which is assumed to be extracted from the input microblog post set be MB_{tf}=*{("church", 2), ("catholic", 2), ("abortion", 2), ("health", 1)}*. Let there be only two articles on Wikipedia which are "Christianity and abortion" and "Obamacare". Let the sets representing these articles with the words and their corresponding *tf-idf* values be {("abortion", 0.76), ("church", 0.68), ("health", 0.23), ("catholic", 0.55)} and {("health", 0.93), ("obamacare", 1.0), ("barack", 0.9)} respectively. Thus, set W_{tfidf} is W_{tfidf}=*{{("abortion", 0.76), ("church", 0.68), ("health", 0.23), ("catholic", 0.55)}, {("health", 0.93), ("obamacare", 1.0), ("barack", 0.9)}* }

In this simple environment, once the formulas are applied, Top=*topics*(Pub_{idf}, MB_{tf}, W_{tfidf}) is computed as

$$Top = \{(\{(\text{“}health\text{”}, 0.93), (\text{“}obamacare\text{”}, 1.0), (\text{“}barack\text{”}, 0.9)\}, 0.23),$$
$$(\{(\text{“}abortion\text{”}, 0.76), (\text{“}church\text{”}, 0.68), (\text{“}health\text{”}, 0.23), (\text{“}catholic\text{”}, 0.55)\}, 0.94)\}$$

The computation gives a relevancy score for the representative set of "Christianity and abortion" as 0.94 and that of "Obamacare" as 0.23. As observed from the set MB_{tf}, this is expected since its content is more similar to the representative set of the "Christianity and abortion" article.

The algorithm of the operations is given in Figure 4.2.

4.2. Prototype

A prototype of the approach is implemented. The prototype uses Phirehose [95] to retrieve posts from Twitter. The prototype that is implemented is similar to that in the algorithm in Figure 4.2. PosgreSQL database management system is used to store the tokens and *tf-idf* scores of each article. For each article, the tokens and their *tf-idf* scores are

Input: Pub_{idf},MB_{tf},W_{tfidf}

Output: Top ▷ A set of Topics (Wikipedia articles) and their similarity scores.

define empty set F ▷ (each element in F is in (t,r) form)

for each (t_1, n) in MB_{tf} **do** ▷ in all input microblog post set tokens and frequencies

  for each (t_2, p) in Pub_{idf} **do** ▷ in all microblog *idf* set tokens and *idf* values

    if $t_1 = t_2$ **then**

    add (t_1,n.p) to F ▷ compute *tf-idf* value for the token and add to F

    end if

  end for

end for

Top $\leftarrow \varnothing$ ▷ empty the set Top

for each d in W_{tfidf} **do** ▷ for each candidate topic (article)

  $x, y, z \leftarrow 0$

  for each (t_1, p) in d **do**

    for each (t_2, r) in F **do**

      if $t_1 = t_2$ **then**

      $x \leftarrow x + p.r$

      end if

    end for

    $y \leftarrow y + p * p$

  end for

  for each (t,r) in F **do**

    $z \leftarrow z + r.r$

  end for

  add (d,$\frac{x}{\sqrt{y}*\sqrt{z}}$) to Top ▷ compute article's cosine similarity with F and add to Top

end for

return Top

Figure 4.2. BOUN-TI algorithm, which identifies topics (Wikipedia article titles) from a microblog post set.

compared with the tokens and *tf-idf* scores of the input microblog post set. The score is saved, then the next article is computed. This approach was not feasible because there were over four million Wikipedia articles at the time of implementation. The large number of Wikipedia articles led to long operations. Comparing an input microblog post set of two minutes with all Wikipedia articles took about 2.5 hours which makes the whole computation over nine days long (see Section 4.3.1 for information about the size of the dataset). The scores of these operations were inspected, and it was observed that most of them were low(< 0.1) due to articles being unrelated to the input post set, but a few (> 0.1) are somewhat related. Since most of the Wikipedia articles are unrelated, pre-filtering Wikipedia articles and comparing only a subset that are likely to be related would speed up comparison.

To overcome this issue, some of the articles were filtered out. From the input post set words (set MB_{tf}), top μ tokens have been chosen according to their *tf-idf* values (r in $\alpha(Pub_{idf}, MB_{tf})$). Pages which do not include any of these tokens are filtered out. To achieve this all Wikipedia article contents are first indexed in Solr. Then, Solr [92] is queried for the selected tokens. This way, Solr returns a subset of Wikipedia articles that are related to distinctive words (words with higher *tf-idf* scores) in the input microblog post set. The implemented algorithm computes the cosine similarities for only the Wikipedia pages returned by Solr.

It may be interpreted from the formal explanations in the approach section that once a Wikipedia article is transformed into tokens and their *tf-idf* value pair sets, the set loses reference to the original article. However, in the implementation this is avoided. As operations are carried out, the algorithm keeps track of which $d \in W_{tfidf}$ corresponds to which specific Wikipedia article.

A Wikipedia dump was taken on August 5, 2013 and was used as the external source. Synsets in WordNet 3.1 [71] were used to enrich the Wikipedia topics. WordNet is a lexical database of English words and phrases. WordNet has over 115 thousand synsets that each correspond to a unique meaning. Each element of a group has the same meaning. Wordnet synonyms are used to improve the chances of detecting a relationship between a topic and a microblog post set. To achieve this, words in the Wikipedia article titles are queried in WordNet. In order to select the relevant synset for a query term, each synset is compared

against the words and the *tf-idf* values of the Wikipedia article using cosine similarity. All words in the selected synset are inserted into the corresponding set d of the article in W_{tfidf}. The value for the words are set as the same as that of the query term. This assigns the words equal importance with the term used to query WordNet. Wordnet output is parsed to reach words as an element of objects in the implemented program. Wordnet provides a command line tool "wn" [98], which allows querying synsets. In the implementation, the output of this tool is parsed and used.

In all computations a stopword list is used [99]. In addition to conventional text processing stopwords, the stopwords in Table A.1 are used. In all sets given in the descriptions, these stopwords were not represented. When the posts and articles were processed, these stopwords were removed.

A post set is retrieved via Twitter API for computing *idf* scores of words in a microblog environment ("Microblog idf store" (set Pub_{idf}) in Figure 4.1). The post set consists of 7,347,669 microblog posts starting from July 11, 2013, 08:57am (GMT) and ending July 16, 2013, 03:30am (GMT). The microblog posts are retrieved by setting the parameters of the Twitter Public Stream sample API [97] endpoint to give a sample of English microblog posts.

The values of words of the collected post set are inspected to gain insight about the public stream. In the post set, the most frequent word is "rt". This is a common word that indicates that the tweet is a retweet. This word exists in 30% of the microblog posts. Recently, Twitter API began giving additional meta information indicating whether a tweet is a retweet of another tweet or not, which makes checking "rt" in a tweet text less important. Some of the most common words like "the" (%24), "to" (%22), "and" (%14), and "not" (%4) are already in the stopwords list. Some of the other most common words were pronoun related terms like "you" (%22), "my" (%12), "me" (%10), "i'm" (%7), and "your" (%5). Observation of the public post set showed that there is not a dominating term or phrase in the post set which makes the set suitable for computing *idf* scores of words in microblogs.

In tokenization of the articles and the post sets, consecutive alphabetic characters were selected as tokens. In microblogs, people use hashtags with camel cases such as #ThisIsA-HashTagWithCamelCase. These camel cases are resolved by considering each word that

starts with a capital letter as a token (i.e.. “this is a hash tag with camel case”).

All Wikipedia articles were preprocessed (i.e.. the texts were tokenized and the stopwords were eliminated), indexed and stored. The data was made ready for use by the implemented system (Set W_{tfidf}). Indexing word frequencies of Wikipedia with PostgreSQL database management system took 41 gigabytes. Indexing how many articles a word exists in on all of Wikipedia took 408 megabytes, and indexing the number of microblog posts a word exists in for all words in microblogs took 199 megabytes of disk space (used to create the set Pub_{idf}). The Solr index took 21 gigabytes of disk space.

The average length of a Wikipedia article is 605 words [100]. Therefore, it is safe to assume that the average number of unique tokens in a Wikipedia article is equal to or below 605. The average number of unique tokens in the experiments with the post sets is 8.2K, which is shown in the following sections. Thus, the cosine similarity operation is applied on vector sizes of average 9K, which is feasible with a desktop computer.

4.3. Experiments and Results

During the presidential and vice presidential debates of the 2012 United States election campaigns, microblog post sets were retrieved from Twitter and then experiments were performed on the post sets. Twitter was very heavily used during these debates [101–104]. The debates were broadcast on television; they were moderated and times were limited for the speakers. In these limited time intervals, various topics were discussed that were agreed upon by both sides before the debate. However, the people who were following the debates were tweeting about any subject. Campaign teams try to communicate particular ideas to people. Therefore, inspecting which topics affect people is important for them, as are the unexpected topics that emerge.

An interval of minutes is referred to using $[m_1,m_2)$ to indicate a time interval beginning with m_1 until m_2 but without including m_2. $m_1 < m_2$.

First, the post sets of the debates are introduced, which are used for evaluation and analysis. Next, the experiments are performed and the results obtained are presented.

4.3.1. Datasets

The three presidential debates' and the vice presidential debate's post sets were retrieved using Twitter's public stream API, employing filter endpoint [105]. To filter tweets, the filter endpoint requires a set of keywords. It returns tweets that include at least one of the keywords in the set. The keywords {"obama", "romney", "barack", "mitt", "republican", "democrat", "elections2012"} were tracked in the first, second and third presidential debates. The keywords {"joe", "paul", "biden", "ryan", "vpdebate"} were added to track the vice presidential debate. Table 4.1 gives post sets and features. Throughout each debate, microblog posts were continuously retrieved from the Twitter stream with these keywords. Tweets associated with each debate were partitioned into two minute intervals resulting in 45 intervals per debate. The opponents were given two minutes to answer the questions from the moderators. Since this is the only information available about the duration of the topics, the data is inspected with this information. However, as will be explained in the following sections, some topics were sustained and are observed in multiple intervals. Each interval is identified as [0-2), [2,4), .. [88,90).

Table 4.1. Characteristics of presidential debates (PD) and the vice PD post sets.

Data label	**Start and end time** (GMT)	**tweet** #	**user** #	**token** #	**unique token** #
1st PD	Oct 04, 2012 02:00:00 Oct 04, 2012 03:29:59	269,990	222,261	2,035,180	149,691
Vice PD	Oct 12, 2012 02:00:00 Oct 12, 2012 03:29:59	270,003	181,854	2,132,895	114,285
2nd PD	Oct 17, 2012 02:00:00 Oct 17, 2012 03:29:59	269,970	222,300	2,057,734	141,905
3rd PD	Oct 23, 2012 02:00:00 Oct 23, 2012 03:29:59	270,018	202,340	2,217,658	128,599

4.3.2. Experiments

Using the retrieved posts for each two-minute interval, a microblog post set was formed. For each debate, 45 post sets were obtained, with 180 post sets in total. The topics of a 90-minute debate (45 sets) could be identified in about 40 minutes by a Pentium-IV 3.2 GHz. computer with 4 GB of RAM.

Human evaluators annotated the topics, which are the Wikipedia pages returned by the system. Topics were annotated according to whether or not they were relevant to the input microblog post set. An annotation interface was designed and prototyped to allow for annotation. The interface shows a representative word cloud for each microblog post set and presents three topic rankings to the evaluators. The threshold μ was taken as $\mu = 20$, $\mu = 50$ and $\mu = 100$ for each ranking in order to investigate the effect of the pre-filtering of topics using Solr. Section 4.2 describes the μ parameter. Each list includes the top ten scored topics.

An evaluation interface view is given in Figure 4.3. Topics listed at the top were annotated by the evaluators as relevant or irrelevant to the microblog post set. Each post set has 6K microblog posts. With this size, it is not easy for an evaluator to go over each post. A word cloud is provided for each post for evaluators to get a general idea of the post set. "See Tweets" link is used by the evaluators to read posts that resulted in these topics. A keyword-based search tool is provided in this interface for evaluators to see tweets of interest using a keyword search. Wikipedia article contents can be explored by clicking on the topic.

Thirty tweet sets were randomly selected from the 180 tweet sets of the four debates. The distribution of these thirty tweet sets across time and debates are shown in Table 4.2.

If the results are inspected, "Barack Obama" and "Mitt Romney" Wikipedia page titles were top scored titles. This is expected, because the datasets were obtained with the words that appear in these pages. Since these topics are expected and related to all post sets, they were removed from the results. The related topics which have titles that include the keywords of the post sets and their plural forms are also removed from the results, since these topics are related to all post sets and they are not distinguished among the intervals. The query

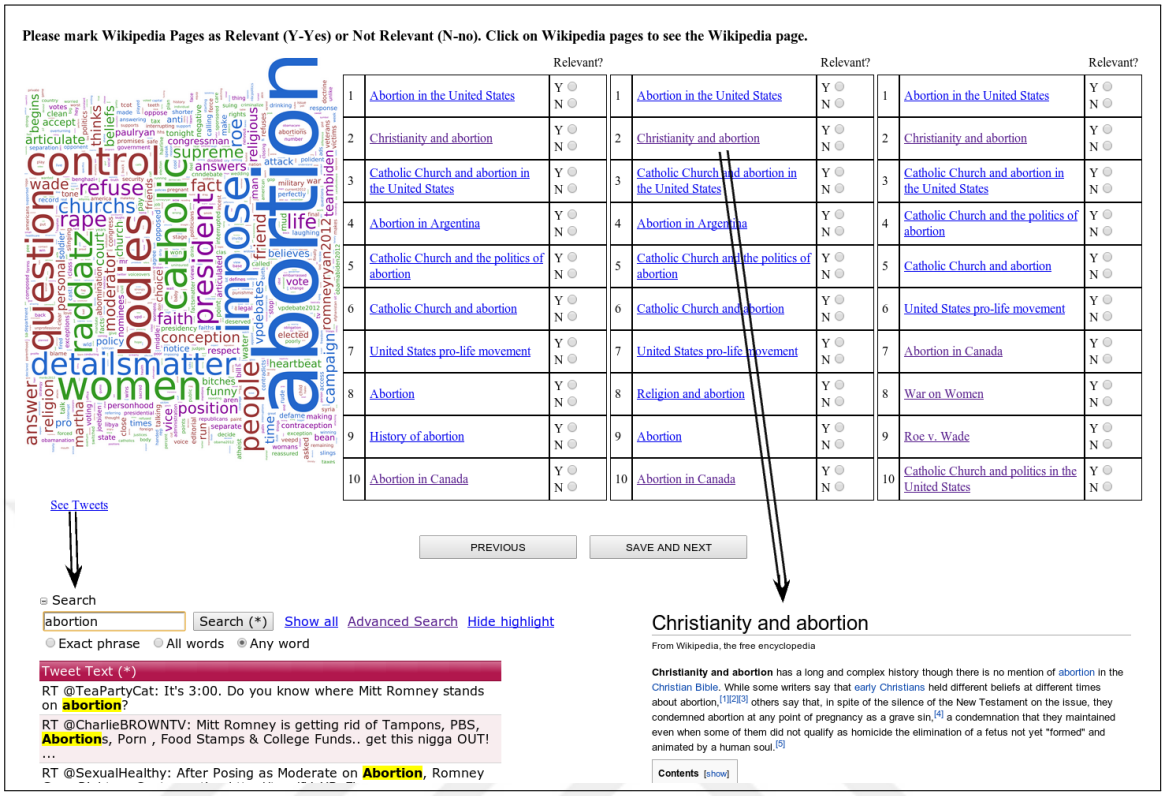


Figure 4.3. The Boun-TI evaluation interface.

words and their plural forms are also removed from the word clouds that are shown in the annotation interface.

After topics were obtained and the annotation interface made ready, two evaluators annotated the topics. Each evaluator was shown 20 microblog post sets and their corresponding results. Ten of these post sets were the same for both evaluators in order to calculate the inter-annotator agreement rate.

4.3.3. Evaluation results

Precision scores that were obtained for the topmost ϕ topics returned by Boun-TI were used as the evaluation metric. ϕ was set as 1, 5, and 10. The precision for each tweet set was computed as the ratio of the number of true positives over ϕ (i.e. the number of all topics that were shown to the annotators). The results achieved are shown in Table 4.3.

Table 4.2. Randomly selected time intervals of the debates

Debate	**Time intervals**
1st PD	[18,20), [26,28), [30,32), [40,42), [50,52), [54,56), [84,86)
Vice PD	[8,10), [16,18), [30,32), [36,38), [68,70), [80,82), [84,86), [88,90)
2nd PD	[26,28), [32,34), [44,46), [50,52), [56,58), [60,62), [74,76), [88,90)
3rd PD	[18,20), [20,22), [26,28), [38,40), [40,42), [48,50), [82,84)

Table 4.3. The precision scores of the top 1^{st}, 5^{th}, and 10^{th} topics according to μ parameter.

	$\mu = 20$	$\mu = 50$	$\mu = 100$
($\phi = 1$) top 1	0.90	0.90	0.96
($\phi = 5$) top 5	0.86	0.89	0.89
($\phi = 10$) top 10	0.82	0.80	0.83

Finally, inter-annotator agreement, (F_1) measures are given in Table 4.4. F_1 measures are calculated as given by Hripcsak and Rothschild [106], who propose a calculation when the annotators cannot annotate all results. Equation 4.8 gives the calculation of F_1 where a is the number of items that both annotators give positive annotation, b is the number of items that the first annotator gives positive while the second annotator gives negative annotation, and c is the number of items that the second annotator gives positive while the first annotator gives negative annotation. There is also a d which is the number of items that both annotators give negative annotation. The article shows that if d is known to be very large, the inter-annotator agreement approaches F_1 given in Equation 4.8.

$$F_1 = \frac{2a}{2a + b + c} \tag{4.8}$$

In this experiment, d is known to be very large, because there are over four million articles to annotate (most of which are unrelated as explained in Section 4.2). It is not likely

Table 4.4. The F_1 measure for the inter-annotator agreement for the 1^{st}, 5^{th}, and 10^{th} topmost topics according to the μ parameter.

	$\mu = 20$	$\mu = 50$	$\mu = 100$
$(\phi = 1)$ top 1	1.00	1.00	1.00
$(\phi = 5)$ top 5	0.96	0.95	0.93
$(\phi = 10)$ top 10	0.93	0.91	0.89

that a significant proportion of the Wikipedia articles would be related to the topics in a given set of microblog posts, especially when the posts are collected based on a time-bound query that constrains the result set. Therefore, the F_1 measure in Equation 4.8 is deemed suitable for the purposes of this work.

An evaluation of the topics revealed that the best results were obtained for $\mu = 100$. with a precision of 0.96. The precision scores decrease as ϕ increases. The top ranked topics are considered to best represent the tweet set. The inter-annotator agreement is consistent with this expectation, which also decreases as ϕ increases. The inter-annotator agreement is higher for the topmost ranked topics.

4.4. Examining Topics over Time

In order to form a better understanding of the identified topics, consecutive sets of topics identified during the 2016 US presidential election debates are examined. Each set corresponds to a two minute interval of one of the four debates. Some of the dominant topics are shown in Figure 4.4. This heatmap is obtained by applying the cosine similarity to all intervals of the debates for chosen topics. A topic is chosen if it is one of the top ranking topics in any of the debates or if it demonstrates a difference between debates. The basic idea is to show interesting topics over the course of the debates.

The heatmap shows that some topics made a strong impression on Twitter. The topics of “Tax”, “Unemployment in the United States”, and “Patient Protection and Affordable Care Act” were more dominant during the first presidential debate. Furthermore, a couple

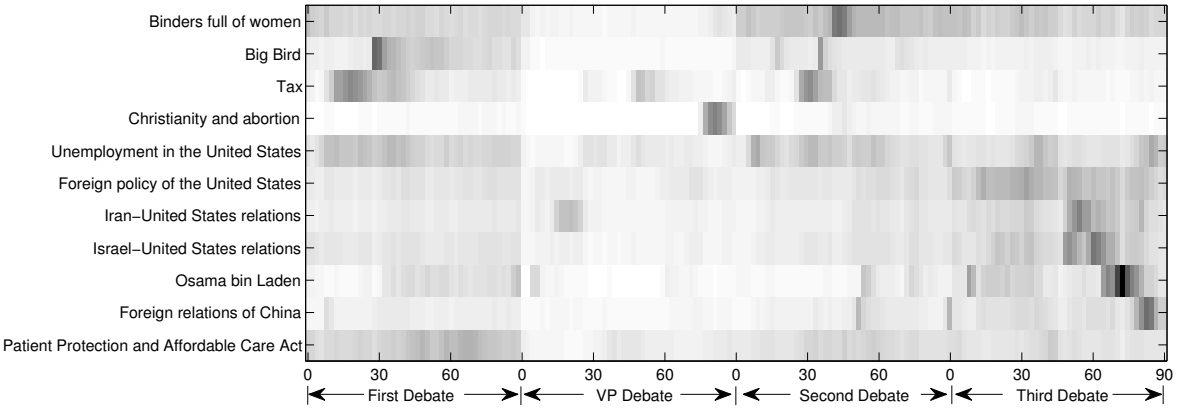


Figure 4.4. The heatmap of topics identified from the tweets gathered during the four debates of the 2012 US elections. x axis shows the minutes of debates and y axis shows a selection of topics. Darker color indicates the topic is talked about more in those minutes.

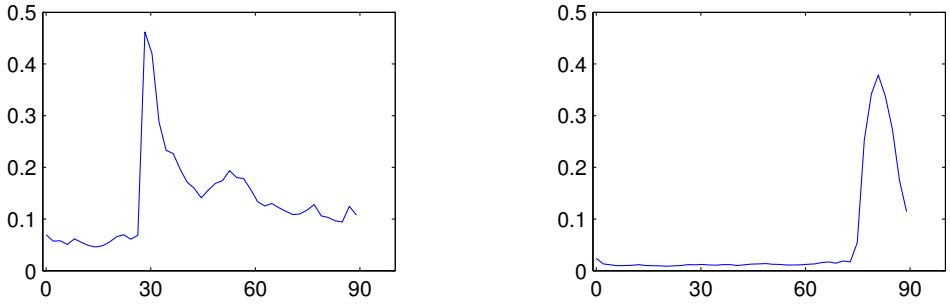
of statements by Mitt Romney regarding "Big Bird" and "Binders full of women" generated quite a significant response. The vice presidential debate did not get as much reaction in general, with the exception of "Christianity and abortion" which was significant since both VP candidates were Catholic, and abortion is a big issue within that faith. The topics of "Tax" and "Iran-United States relations" also emerged, but to a lesser degree. The topic of "Binders full of women" remained a topic of discussion in the second and third presidential debates, demonstrating the significance given to women's issues by Twitter users. The third debate also generated "Tax" and "Unemployment" topics. The third presidential debate generated far more interest in foreign policy issues as well as "Osama bin Laden". The debates were announced with the following themes: (1) "Domestic policy" for the first presidential debate, (2) "Foreign and domestic topics" for the vice presidential debate, (3) "Foreign and domestic issues" for the second presidential debate, and (4) "Foreign policy" for the third and final presidential debate. It is interesting to examine how the topics that were identified on Twitter relate to the formally scheduled topics.

Some topics appear very strong – such as "Big Bird" and "Christianity and abortion" - indeed indicating a strong public reaction. The term "Big Bird" was referred to by Mitt Romney during the first presidential debate, when he promised to cut the subsidy paid to the Public Broadcasting Service (PBS) if he were elected. He said that he loved the popular PBS television program character "Big Bird", but he did not think government should support

it. His statements caused a strong reaction on social media, which led to a sharp increase in the incidence of the terms “Big” and “Bird”. While these terms occurred but once during the televised debate, the impact on social media endured far longer. The difference in the occurrence of these terms is shown in Figure 4.5 (a), where around the 28^{th} minute of the first debate the term spikes.

In order to relate the televised debate to reactions on Twitter, resources that transcribed the debates were used. There was no ideal transcription to relate to the highly temporal and instant tweet tempo, however there are sufficiently useful transcriptions. The transcripts published by The New York Times and CNN News websites were utilized to relate what was said during the debate to what was being posted on Twitter. The New York Times website [107] published the transcripts of the debates with the minutes and the seconds for each comment from the beginning of the debate (i.e. 0:02:44; President Obama; Well, thank you very much, Jim, for this opportunity). The CNN website [108], on the other hand, provides the timestamps of the speeches but does not provide them as frequently as the New York Times website. These two sources of information are aligned to provide a reference to the actual debate content. The New York Times documents that Mitt Romney said “Big Bird” in the 26^{th} minute (4-5 seconds after the $1,564^{\text{th}}$ second). According to the CNN website, the debate started 100 seconds after Oct 04, 2012 02:00:00 (GMT). Therefore, just a few seconds after Mitt Romney uttered his line about “Big Bird”, a huge response is observed on Twitter. Interestingly, during the second debate, this topic emerged during the 15^{th} and 40^{th} minutes, indicating that the topic had not died off in the public arena. The debate transcripts have no reference to this topic as it was not mentioned during the debate. It is this type of information that is very useful to capture when trying to track public interest. The public will talk about what interests them. Being able to track which messages resonate with the public and which performances (public speeches, concerts, warnings) best reach them is important.

In the second presidential debate, Mitt Romney used the phrase “Binders full of women”. Microblog users reacted to this quickly. The same topic is also observed in the third presidential debate, as shown in Figure 4.4.



(a) (b)

Figure 4.5. The "Big Bird" and "Christianity and Abortion" topics and how their scores varied over time during the first PD and the Vice PD respectively. The y axis is the score of the topic and x axis is the time in minutes. (a) "Big Bird" topic in the first presidential debate. (b) "Christianity and abortion" topic in the vice presidential debate.

"Unemployment in the United States" and "Tax" show similar behavior, except in the beginning of the second debate. A reason for this may be that, in the early parts of the debate, the opponents talked about unemployment issues but not explicitly about tax issues. The opponents mostly talked about health care related topics and Obamacare in the first debate. "Patient Protection and Affordable Care Act", also known as "Obamacare" has higher scores in this debate.

The third debate was about foreign policy issues. The effect is seen on the topics of the third debate. The "Foreign policy of the United States" received higher scores in this debate. It is observed that reasonable results are produced by BOUN-TI. For example, the second half of the third debate can be tracked from the heatmap. The topics in those time intervals were in order: "Israel-United States relations", "Iran-United States relations", "Israel-United States relations" (again), "Osama bin Laden", and "Foreign relations of China". These topics received the highest scores in the minutes the presidential candidates were talking about them. When the "Iran-United States relations" is ranked first, the second topic was "Views on the nuclear program of Iran". This was also a related topic. Similar behavior by these two topics was also observed in the vice presidential debate. The transcriptions confirm these results.

During the vice presidential debate, the moderator asked the vice presidential candidates (Vice President Joe Biden and Congressman Paul Ryan) about how their Catholic faith informed their political stances. Twitter users reacted to this topic immediately. The highest rated topic, "Christianity and Abortion" can be seen in Figure 4.5b) at the 74^{th} minute. The topics that followed in rank were "Abortion in the United States", "Catholic Church and abortion in the United States", and "Catholic Church and abortion" – all related to the main topic.

It is worth noting that the Twitter users do not specifically use the terms "Christianity and abortion" and "Reactions to the death of Osama Bin Laden", which are high level topic descriptions. Instead, they use lower level terms that describe what specifically is on their minds, for example "catholic", "church", and "abortion". The resulting topics include the occurrences of these terms, leading to the most relevant titles. Even thought their style and format are very different, the fact that both sources are created with user generated content has a beneficial influence both in locating relevant topics as well as ranking them. If the titles of the articles had been used to identify topics, then topics such as "Osama Bin Laden", "Catholic Church", and "Abortion" would have resulted. This has been confirmed by inspecting the terms in the post sets with the highest *tf-idf* scores.

4.5. Identifying topics for single versus multiple microblog posts

The approach proposed in this work treats a set of microblog posts as a unit. It treats all posts as a document, so to speak. An alternative approach might have been to process each microblog post as a unit (a document) and then to aggregate the topics resulting from each post to yield topics for a collection. To gain some insight into such an alternative, each post in the set was given to the TagMe linker, with the results related to Wikipedia pages. Detailed information about TagMe is given in Section 3.5. The most frequently occurring topics were considered the topics for that set and compared with the BOUN-TI topics. Tables 4.5 and 4.6 show the top five topics for these two approaches corresponding to the [28,30) minutes of the first presidential debate and [80,82) minutes of the vice presidential debate respectively.

Both approaches resulted in "Big Bird" as the top ranked topic for the [28,30) minutes of the first debate. The same approach was applied to each microblog post set, which resulted

Table 4.5. Comparison of BOUN-TI topics and topics from aggregating TagMe linkings. The topics are obtained from [28,30) minutes of the first presidential debate

Rank	**TagMe**	**Boun-TI**
1	Big Bird	Big Bird
2	Lava	Bush tax cuts
3	(F word)	Economic policy of the George W. Bush administration
4	PBS	Tax Relief, Unemployment Insurance Reauthorization, Job Creation Act of 2010
5	You (Time Person of the Year)	United States presidential election, 2012

Table 4.6. Comparison of BOUN-TI topics and topics from aggregating TagMe linkings. The topics are obtained from [80,82) minutes of the vice presidential debate

Rank	**TagMe**	**Boun-TI**
1	Abortion	Christianity and abortion
2	Catholic Church	Abortion in the United States
3	Belief	Catholic Church and abortion in the United States
4	Transmitter	Catholic Church and abortion
5	People	Abortion in Argentina

in this Wikipedia article being ranked first. The fourth topic of TagMe, "PBS", is captured by BOUN-TI. "PBS" is mainly mentioned in the context of "Big Bird". Other topics of TagMe in these results are not relevant to the topics talked about during those times. All topics of BOUN-TI were relevant to the post set. These topics were identified due to the similarities between the words in the post set and Wikipedia pages.

Both approaches returned related results for the [80,82) minutes of the vice presidential debate. Aggregation of TagMe results gave separate articles such as "Abortion" and "Catholic Church" BOUN-TI on the other hand returned inclusive and descriptive results such as "Christianity and Abortion" and "Abortion in the United States". The nature of

BOUN-TI leads to these results. BOUN-TI considers all words in the post set, unlike TagMe which considers the words in the same post for identification. In the top five results of BOUN-TI, "Abortion in Argentina" appears, but it is not related to the post set. This false positive ranked high because the term "abortion" frequently appears in that article.

TagMe provides confidence values for linkings. The values show how reliable a surface form is (p) and how reliable the linking is (ρ). The initial results did not consider these values. Table 4.7 gives the TagMe topics if the entities were considered above thresholds $p > 0.4$ and $\rho > 0.15$. The aim of this experiment is to see if the increasing threshold affects the identified topics which were set as zero before. For the [28-30) minutes of the first presidential debate, "PBS" and "Big Bird" were already the top scored topics. In the increased threshold listing, another related topic, "Sesame Street", gets into the top five. The other two are unrelated. For the [80-82) minutes of the vice presidential debate, the top four are related to the topics in the posts. In the initial run, there were two unrelated topics, while in these results there is only one unrelated topic. The quality of related topics also increased, according to the observations.

Table 4.7. Topics obtained by aggregating TagMe linkings that are extracted from [28,30) minutes of the first presidential debate, and [80,82) minutes of the vice presidential debate for $p > 0.4$ and $\rho > 0.15$.

Rank	**[28,30) minutes of the first presidential debate**	**[80,82) minutes of the vice presidential debate**
1	Roe v. Wade	PBS
2	Abortion	Big Bird
3	Anti-abortion movements	Dot (diacritic)
4	Incest	Sesame Street
5	At or with Me	LMFAO

Although there is an increase in the quality and quantity of the identified topics of aggregation of TagMe linkings, increasing the threshold could not capture the topics BOUN-TI captures.

4.6. Conclusions

While working with BOUN-TI, interesting topics that represent the posts were found. Topics with multiple concepts and entities such as "Christianity and Abortion" and "United States-Iran Relations" represent the relationships among terms and concepts, and are interesting topics which are unlikely to be identified if comparison to Wikipedia articles is not utilized for topic identification.

There were some issues with the topics identified by BOUN-TI, however. The main problem is related to crosstalk. Unrelated topics in the post set may be merged and identified as one topic. For example, when the issues that are talked about in the post set are "coal in Canada" and "economy policy in the United States", the topic "Economy of Canada" would be identified as a topic. This identification is incorrect. The problem of crosstalk is therefore identified as an area to be addressed in the future development of BOUN-TI. One solution is to identify elements of topics and their relationships so that topics with unrelated elements can be eliminated. S-BOUN-TI, which is introduced in Chapter 6, relates elements of topics. The results of S-BOUN-TI can provide relationships among elements to improve the results of BOUN-TI.

Another problem was with incorrect year referrals in topics. Some of the topics of BOUN-TI include year referrals to past years such as "United States presidential election debates, 2008". Special processing of year referrals was needed to solve this problem. In the further development of BOUN-TI, the year referral problem was solved by filtering out the topics which have a year referral that is not referred to in the post set in a certain amount of posts.

The BOUN-TI prototype considers Wikipedia as a static encyclopedic source. However, Wikipedia updates should be checked in order to identify recent topics. Not all articles are updated regularly on Wikipedia, but there are articles that are immediately updated, such as the page of a celebrity when he or she dies, or when a match between two major sports teams is finished. Updates can be used for validating identified topics. New articles can also be tracked, and in the case of similar content found in microblogs, these articles can be output as topics.

BOUN-TI aims to identify the main topics of a post set, which are then represented by Wikipedia page titles. Subtopic identification is also a problem, which may be represented with subsections in a Wikipedia page. This problem has not been addressed and is a future direction. Sometimes, topics that are subsections of a Wikipedia page may have a corresponding independent page that is interesting to observe. If the post set content focuses on the subtopic, cosine similarity will give higher similarity scores for the corresponding page in this case.

In the early stages of work on BOUN-TI, only the extended abstracts [109] of Wikipedia pages were considered as Wikipedia page content. The extended abstract is the body of a Wikipedia page starting from the beginning of the page until the first heading. The reason behind this decision was that an extended abstract is the summary of the whole article, which seems to imply that the use of only those words in the extended abstract may be sufficient for a topic model of the article. Interestingly, working with only the extended abstracts resulted in lower quality topics than when the whole article was considered. Extended abstracts include more abstract terms and phrases than the rest of page, but in the short texts, like tweets, people usually do not use abstract terms and phrases, but rather specific terms and phrases. These specific terms and phrases may be repeated several times in the rest of a Wikipedia page, making their *tf-idf* values higher for that page. Thus, considering the whole content of Wikipedia articles improves identification when the content is to be compared to that of microblog posts.

Evaluation is a challenge in this domain, since it is difficult to manually annotate huge numbers of microblog posts. A manually annotated post set has been obtained that could be used for further research and evaluation.

Considering the evaluation results and observations in this chapter, it can be concluded that the proposed approach achieves promising results, identifying topics effectively, and is suitable for further research and study.

5. TOPICO: AN ONTOLOGY FOR MICROBLOG TOPICS

Topic$_{\mathcal{O}}$ is designed to represent topics that are posted on microblogs. Microblog posts are highly temporal and usually refer to the present moment, although they may refer to the past or future. People typically post about current events like sports, news, politics and entertainment. Posts often refer to people, some of whom are well-known. They also refer to places, either related to the topics of the posts or where the post was posted from. This chapter describes the classes and properties of *Topic*$_{\mathcal{O}}$, using examples that highlight the characteristics of microblog posts.

Topic$_{\mathcal{O}}$ defines concepts to express "agents", "places", and "temporal" aspects related to topics. Furthermore, it is designed to refer to data in Linked Open Data in order to capture relevant aspects of a topic. In the prototype developed for extracting topics (Chapter 6), the Linked Data referred to most often is from DBpedia.

An overview of the classes, data properties, and object properties of *Topic*$_{\mathcal{O}}$ are shown in Figure 5.1. The object properties of *Topic*$_{\mathcal{O}}$ are shown in Figure 5.2. In further chapters the prefix "topico:" is used to refer to the *Topic*$_{\mathcal{O}}$ namespace.

Whenever possible, *Topic*$_{\mathcal{O}}$ utilizes existing ontologies and vocabularies (see Section 3.7). The "Friend of a Friend" (FOAF) ontology is used to refer to agents. "Basic Geo (WGS84 lat/long) Vocabulary", "GeoNames Ontology", "DBpedia ontology", and "schema.org vocabulary" are used to describe locations. "W3C OWL-Time ontology" is used for temporal expressions.

The "topico:Topic" class represents topics. *Topic*$_{\mathcal{O}}$ uses four main relations to describe topics: "topico:hasAgent", "topico:hasLocation", "topico:hasTemporalExpression", and "topico:isAbout". All elements other than agents, locations, and temporal expressions are specified with the "topico:isAbout" property. Additionally, *Topic*$_{\mathcal{O}}$ defines several object and data properties to describe meta characteristics about the topic. The details are defined in sections 5.1, 5.2, and 5.3.

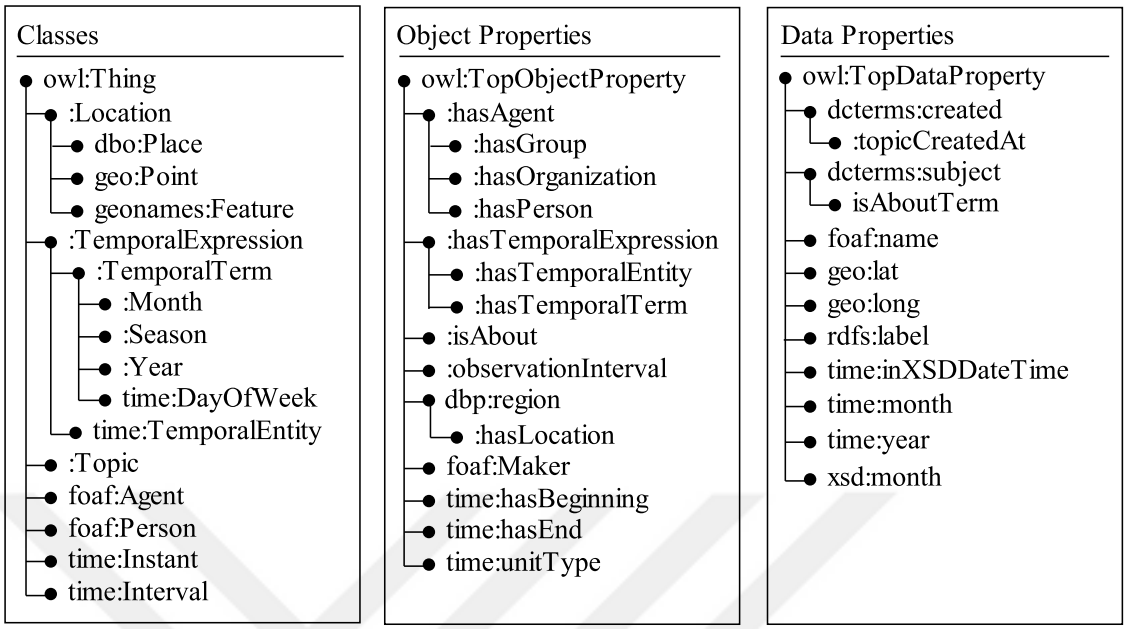


Figure 5.1. Overview of $Topic_{\mathcal{O}}$. The definition of prefixes are found in Section 3.8. Items that start with ":" are defined by $Topic_{\mathcal{O}}$.

Essentially, a topic consists of agents, locations, temporal information, related elements, and meta-information. Figure 5.3 shows an example instance of "topico:Topic". The topic is related to the presidential candidates "dbr:Donald_Trump" and "dbr:Hillary_Clinton" with the "topico:hasPerson" property. It is related to "dbr:United_States_presidential_election_debates", and "dbr:Racism" with the "topico:isAbout" property. The IRI "http://ex.org" is used as the domain of individuals to separate individuals from the ontology domain and shorten the URI to fit the page. Figure 5.4 shows an example of a topic related to two posts. The resource "dbr:United_States_presidential_election_debates" is related to this topic via "topico:isAbout". The remainder of this chapter provides details of $Topic_{\mathcal{O}}$.

5.1. Agents

To express agents "foaf:Agent" is used, which has the subclasses "foaf:Person", "foaf:Group", and "foaf:Organization". The "topico:hasAgent" property defines the "foaf:Agent" of "topico:Topic". The inverse relation of "topico:hasAgent" is "topico:isAgentOf". The "topico:hasPerson", "topico:hasGroup", and "topico:hasOrganization" properties are subproperties of "topico:hasAgent", where the ranges of these properties are "foaf:Person", "foaf:Group", and "foaf:Organization" respectively. The properties "topico:isPersonOf",

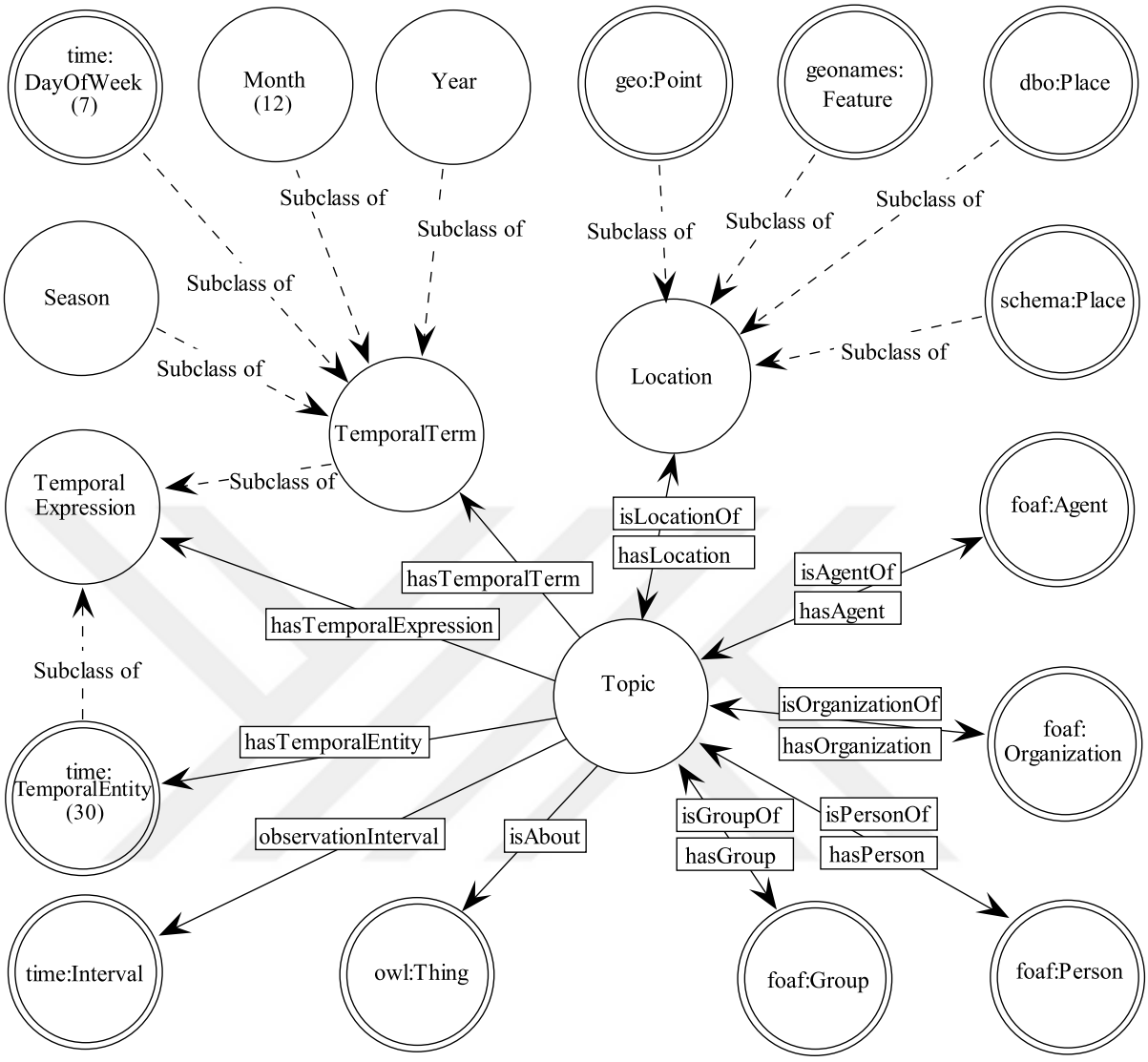


Figure 5.2. Object properties of $Topic_{\mathcal{O}}$. The classes in double circles are external classes, such as FOAF. Numbers in class circles indicate the number of members, i.e. "time:DayOfWeek" has 7 members

"topico:isGroupOf" and "topico:isOrganizationOf" are their respective inverse properties.

An example "topico:Topic" with two agents, one of which is of type organization and the other of type person is shown in Figure 5.5

5.2. Locations

Locations are specified with the "topico:Location" class, which has the subclasses "dbo:Place", "schema:Place", "geonames:Feature", "schema:Place" and "geo:Point" to sup-

```
<owl:NamedIndividual rdf:about="http://ex.org/topics/topic1">
   <rdf:type rdf:resource="http://soslab.cmpe.boun.edu.tr/ontologies/topico.owl#Topic"/>
   <foaf:maker rdf:resource="http://ex.org/resources/TopicPopulationAlgorithmV11"/>
   <rdfs:label xml:lang="en">2016 US elections first presidential debate 0-2/7</rdfs:label>
   <topicCreatedAt
      rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTimeStamp">2017-01-26T11:37:22Z
   </topicCreatedAt>
   <observationInterval rdf:resource="http://ex.org/intervals/interval_1"/>
   <hasPerson rdf:resource="http://dbpedia.org/resource/Hillary_Clinton"/>
   <hasPerson rdf:resource="http://dbpedia.org/resource/Donald_Trump"/>
   <isAbout rdf:resource="http://dbpedia.org/resource/Racism"/>
   <isAbout rdf:resource="http://dbpedia.org/resource/United_States_presidential_election_debates"/>
</owl:NamedIndividual>
```

Figure 5.3. A "topico:Topic" instance regarding Hillary Clinton, Donald Trump and Racism

port a variety of location descriptions. The property "topico:hasLocation" with the domain "topico:Topic" and the range "topico:Location" is used to specify the locations of a topic. Figure 5.6 shows an example topic with a location.

5.3. Temporal expressions

Microblog posts may refer to a time relative to the time of posting (i.e. now, tonight, tomorrow), a duration (i.e. 2 hours), a reference to a proper temporal noun (i.e. Wednesday, August) or a specific date (i.e. 20.Nov.2017). Topics are timestamped, thus both the time of utterance and any temporal expression within the post can be captured. The observation time of a topic is specified with the "time:Interval" instance, which corresponds to the timestamp of the earliest and the latest posts in the post set.

The types of temporal expressions that occur in microblogs may be absolute or relative. Temporal expressions may include proper nouns such as "Monday" or "June". W3C OWL-Time ontology [110] W3C working draft defines many useful temporal expressions. $Topic_{\mathcal{O}}$ uses these definitions, also defining those that W3C OWL-Time ontology does not cover.

The "topico:TemporalExpression" class is defined as the base class for temporal expressions. It has two main subclasses "topico:TemporalTerm" and "time:TemporalEntity". Relative temporal expressions like "now", "tomorrow", and "today" are defined as instances of

post 1: Now Trump is on to stop and frisk and criminalizing immigrants. #debatenight

post 2: Did @realDonaldTrump just push stop and frisk as a way to fight crime and then blame those policies on the Democrats as a bad thing?

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <isAbout rdf:resource="http://dbpedia.org/resource/Stop-and-frisk_in_New_York_City"/>
   ...
</owl:NamedIndividual>
```

Figure 5.4. An example of a topic's "topico:isAbout" property. The upper box shows a collection of two microblog posts. The lower box shows the part of a topic whose value for "topico:isAbout" property is "http://dbpedia.org/resource/Stop-and-frisk_in_New_York_City". The IRI "http://ex.org/topic/1" denotes the id of a topic, which is instantiated via some topic detection system.

post : Pres. Trump calls on DOJ for voter fraud investigation

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <hasOrganization rdf:resource="http://dbpedia.org/resource/United_States_Department_of_Justice"/>
   <hasPerson rdf:resource="http://dbpedia.org/resource/Donald_Trump"/>
   ...
</owl:NamedIndividual>
```

Figure 5.5. A "topico:Topic" instance with agents "dbr:United_States_Department_of_Justice" and "dbr: Donald_Trump". The terms "Trump" and "DOJ" in the post correspond to the DBpedia resources seen in the topic.

"topico:TemporalExpression". For example, "topico:Today" and "topico:Tomorrow". There are 30 such instances (see Table B.1).

The "topico:TemporalTerm" class addresses proper nouns like the day of the week and month. It has the subclasses: "time:DayOfWeek", "topico:Month", "topico:Season", and "topico:Year". Temporal references frequently occur in microblog posts. Each month is represented with an instance such as "topico:January". These instances are mapped to W3C OWL-Time ontology with the object property "time:unitType" set to "time:unitMonth" and the data property "time:month" set to a value of type "xsd:gMonth". This mapping enables interoperability between $Topic_{\mathcal{O}}$ and W3C OWL-Time ontology elements. The "topico:Year" class uses W3C OWL-Time ontology to express years, with the properties

post : Standing Rock: US veterans join North Dakota protests #RTGWorld

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <hasLocation rdf:resource="http://dbpedia.org/resource/North_Dakota"/>
   ...
</owl:NamedIndividual>
```

Figure 5.6. A topic related to the protests in Standing Rock, North Dakota, which specifies the property "topico:hasLocation" defines the location to be "dbr:North_Dakota".

"time:unitType" set to "time:unitYear" and the data property "time:year" set to a value with type "xsd:gYear". The "topico:Season" class has the instances "topico:Summer", "topico:Winter", "topico:Fall", and "topico:Spring" to express seasons. Terms like "Spring festival", "Summer Workshop", "Fall semester" are common in microblog posts and relevant to topics.

The "time:TemporalEntity" class is used to express exact dates and times. Its subclass "time:Instant" specifies dates using one of the seven data properties according to need (i.e. "time:inXSDDate" with range "xsd:dateTime"). The "time:Interval" class is a subclass of "time:TemporalEntity", which is used to express durations such as "two hours" and "ten days".

The "topico:hasTemporalExpression" with the domain "topico:Topic" and range "topico:TemporalExpression" is used to specify temporal aspects of a topic. It has the subproperties "topico:hasTemporalTerm" and "topico:hasTemporalEntity" with the domain "topico:Topic" and the ranges "topico:TemporalTerm" and "time:TemporalEntity" respectively. Figures 5.7 and 5.8 show several examples of temporal terms for topics.

5.4. Meta information

Topics have some meta information, such as when they were created and by whom. Since topics are associated with collections of posts, they are also associated with a related time interval corresponding to the earliest and latest timestamped posts. This interval will determine when the topic became relevant. Microblog posts are highly temporal, with topics

post : RT @selenaworld: Since #Brangelina are divorcing now, here's my favorite couple in the world, they really make me believe in love. https://...

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <hasTemporalExpression rdf:resource="http://soslab.cmpe.boun.edu.tr/ontologies/topico.owl#Now"/>
   ...
</owl:NamedIndividual>
```

Figure 5.7. An example temporal expression. The temporal expression "now" results in the topic having "topico:hasTemporalExpression" set to "topico:Now"

post : 115.2 million people watched Super Bowl XLIX. Here's to hoping more than that are watching tonight. #Debates2016 #debatenight

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <hasTemporalExpression rdf:resource="http://soslab.cmpe.boun.edu.tr/ontologies/topico.owl#Tonight"/>
   <hasTemporalTerm rdf:resource="http://ex.org/resources/year_2016"/>
   ...
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="http://ex.org/resources/year_2016">
   <time:unitType rdf:resource="http://www.w3.org/2006/time#unitYear"/>
   <time:year rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2016</time:year>
</owl:NamedIndividual>
```

Figure 5.8. A topic with two temporal expressions corresponding to "2016" and "Tonight" in a post.

frequently changing. Trending topics represent the most prominent manifestation of these changes. This interval is specified with the "topico:observationInterval" property, an instance of topic observation time. The domain of this property is "topico:Topic" and the range is "time:Interval".

The topic creation time is specified with the "topico:topicCreatedAt" data property, with the domain "topico:Topic" and the range "xsd:dateTime". The topic creator is an instance of "foaf:Agent". The "foaf:maker" property is used to specify the creator of the topic instance, which may represent a software agent. Figure 5.9 shows a topic with an observation interval, a creation date and time, and a maker.

```
<owl:NamedIndividual rdf:about="http://ex.org/topic/1">
   <foaf:maker rdf:resource="http://ex.org/resources/TopicPopulationAlgorithmV11"/>
   <rdfs:label xml:lang="en">2016FirstDebate 20-22 minutes topic 1</rdfs:label>
   <topicCreatedAt rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTimeStamp">2017-01-26T11:53:39Z
                                                                                  </topicCreatedAt>
   <observationInterval rdf:resource="http://ex.org/intervals/interval_1"/>
   ...
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="http://ex.org/intervals/interval_1">
   <rdf:type rdf:resource="http://www.w3.org/2006/time#Interval"/>
   <time:hasBeginning rdf:resource="http://ex.org/instants/instant_1"/>
   <time:hasEnd rdf:resource="http://ex.org/instants/instant_2"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="http://ex.org/instants/instant_1">
   <rdf:type rdf:resource="http://www.w3.org/2006/time#Instant"/>
   <time:inXSDDateTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2016-09-27T01:20:00Z
                                                                              </time:inXSDDateTime>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="http://ex.org/instants/instant_2">
   <rdf:type rdf:resource="http://www.w3.org/2006/time#Instant"/>
   <time:inXSDDateTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2016-09-27T01:21:59Z
                                                                              </time:inXSDDateTime>
</owl:NamedIndividual>
```

Figure 5.9. An example topic with expressing meta information such as the observation interval, creation time, the maker, and given label of the topic. Observation interval is another individual that refers to instant individuals. The remainder of the topic is not shown here (indicated with ...).

5.5. Object and data properties

As mentioned earlier, existing ontologies have been utilized whenever possible. Figure 5.10 shows the inverse properties of the object properties in Figure 5.1. Each inverse object property is a subproperty of "topico:isRelatedToTopic". This results in each element of a topic having the "topico:isRelatedToTopic" property with the topic.

The creation time of a topic is specified with the "topico:topicCreatedAt" property. The terms that are related to the topic but could not be linked to any Semantic Web resource are specified with the "topico:isAboutTerm" property. This enables the use of "dcterms:created" and "dcterms:subject" properties. As such, these $Topic_{\mathcal{O}}$ properties are defined as subproper-

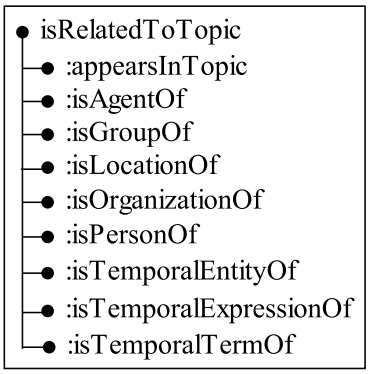


Figure 5.10. Inverse properties of the object properties of $Topic_{\mathcal{O}}$. Each inverse property is a subproperty of "topico:isRelatedToTopic".

ties of the external properties. The ranges are "xsd:datetime" and "rdfs:Literal" respectively. The domain of both is "topico:Topic".

$Topic_{\mathcal{O}}$ is used to express topics that are extracted with an approach suggested in Chapter 6. The generated topics are served via a Fuseki server found at "http://193.140.196.97:3030/datasets.html". Machine interpretable topic expressions are explorable at "http://soslab.cmpe.boun.edu.tr/sbounti".

6. EXTRACTING MACHINE INTERPRETABLE TOPICS

Microblog environments provide a wide variety of topics. Identifying the topics of a collection of posts is interesting because individual people contribute to different parts of a topic. Thus, in a microblog, it is assumed that a topic may have several "elements", which are meaningful parts of a text. A post set can provide many topics. Different microbloggers may talk about the same elements. The elements may be related to one or to more than one topic. For example, there may be different issues discussed regarding one person, such as a politician. This suggests that the politician is related to more than one topic. Or, a post may provide one or more elements of a topic. As an example, Figure 6.1 gives three posts. The first post provides elements "Barack Obama", "middle class", "tax", and "debate". The second post provides elements "Mitt Romney", "Barack Obama", "debate", "Denver", and "2012". The third post provides elements "Mitt Romney", "debate", "Denver", and "2012". In this example, "Barack Obama" is mentioned with "middle class" and "tax" but "Mitt Romney" is not. If it is assumed that there is a relation between two elements if they are obtained from the same post, and if the posts in the figure are considered, the possible topics that can be extracted from this post set would be {Barack Obama, Tax, Middle Class}, and {Barack Obama, Mitt Romney, Denver, Debate, 2012}.

If the elements of a topic are identified and related then they can be expressed by a topic structure (see Figure 6.2 for an illustration). The structure that holds the elements together can be utilized in various ways such as visualization, searching, faceted searching, providing relationships among elements to be utilized by other tasks (such as better identification of the topics in Boun-TI) and issuing complex queries (using one element's existence in multiple topics). An example query is "Who else is mentioned with Hillary Clinton" which can be answered if such a topic structure is available. This query (see Figure 6.10) can be

post 1: @GlobalGrind: FACT: Barack Obama cut taxes for the middle class 21 times. #Debate2012
post 2: Bring it on @MittRomney! @BarackObama is gonna own you of this I have no doubt! #Debate2012 #DebateDenver
post 3: ROMNEY ROMNEY!!!!!!! #debate2012 #denverdebate

Figure 6.1. Several posts in which some elements are related.

answered using the relationships within topics because the topics are assumed to provide the related persons. Another example query is "When did the issues that are most referred to involving Hillary Clinton or Donald Trump emerge". This query is interesting, because political campaign managers may want to know the answer to this query in order to track public responses to issues. Multiple topics may exist that are discussed in the same time interval that include Hillary Clinton but not Donald Trump or vice versa. Therefore, answering this query requires relating topics. The topics that occur during the same time interval should be obtained. Then, from these topics, the issues related to Hillary Clinton, Donald Trump, or both of them should be extracted. Finally, in a proper format, the output should be projected. A sample of these queries (see Figure 6.12) and processing of their results by projecting them to a table (see Figure 6.13) are given in this chapter.

If Linked Data resources are identified as elements of topics, the topics obtained from microblog post sets can be utilized beyond that which is provided by a microblog platform (such as Twitter). For example, one can query topics such as "Show me the rock music artists that people have recently been talking about along with the locations of their concerts". For this query (see Figure 6.15), Linked Data is queried for the genre of rock music artists. The artists and the related locations are provided in the identified topics. These two sources of information should be joined to answer the query.

To identify topics, elements of topics should first be identified. To identify elements, meaningful parts (spots) should be identified. If the spot can be expressed as a Linked Data resource, that resource should also be identified. Luckily, there are state of the art entity linkers (annotators) which work well to some extent. These approaches tailor conventional entity recognition and linking approaches to work with short texts such as microblog posts. It has been observed, however, that there are still problems that need to be solved in this field of research due to the limited context of posts. Examples of these problems are wrong linkings, low confidence linking, or not linking a spot at all while there exists a resource for the spot. Although the approach introduced in this chapter is not an entity linking approach, some of these problems are addressed. The goal is to fix wrong links and link unlinked spots using the information collected from other related posts, which provide a substantially greater context for the task. After identifying elements of topics, the challenge is to decide which elements go into which topics.

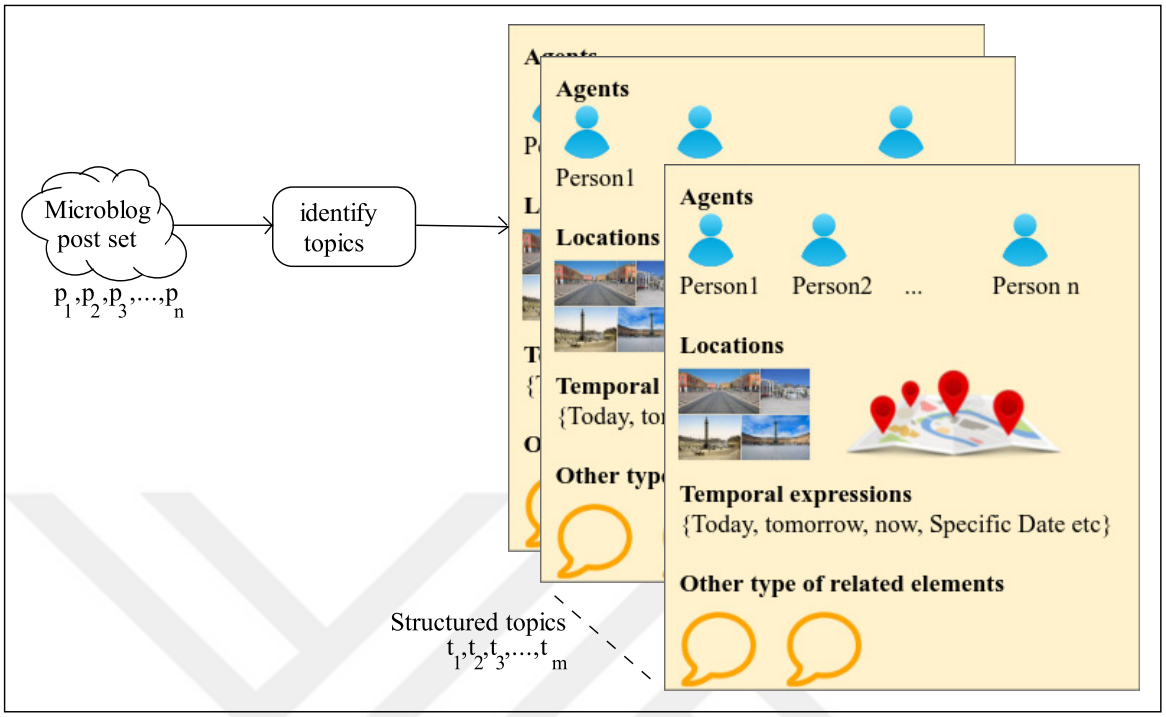


Figure 6.2. Illustration of topic identification. Given posts, related meaningful parts are identified and they are expressed as elements of a formed structure.

This chapter introduces a novel approach, S-BOUN-TI, to overcome these challenges. The idea is to first relate elements that are identified. Since microblog posts are of very limited length, the presence of linked spots (entities) is considered valuable. Users write short and descriptive posts to transfer their ideas. If an element is identified in a single post, it is valuable. If the post contains more than one linked spot it is assumed that there is a relationship between them. Furthermore, the linked entities are considered to be related. The more often the same co-occurrence happens the more trusted that relation is, because it means that numerous users have posted about the same elements. Furthermore, it is assumed that if an element or a relationship (two elements) is identified in a high number of posts, more users agree on that element or that relationship.

Once the elements are related, these relations are represented using a graph structure, where vertices are the elements and the edges among the elements are weighted according to the co-occurrence of their vertices' spots in the same post. Then, the graph is pruned to obtain strong relations and their vertices. This operation outputs the edges and vertices that a higher number of users contribute, which are the relations that a higher number

of users confirm. This is especially important because the approach seeks the topics of a collection of posts by multiple users. Then, from the graph of related elements, the topics are formed. The topics are represented using the expressions in $Topic_{\mathcal{O}}$, which was introduced in Chapter 5. The types of elements are identified by querying Linked Data. Several steps are explained in this chapter which are related to agent, location and temporal expression identification. These steps identify user mentions such as persons and organizations, link to resources in Linked Data, identify locations according to the usage of the spot in the post set, and identify temporal expressions that $Topic_{\mathcal{O}}$ defines as individuals in the ontology, which are the expressions that are most often found in microblog posts.

The rest of this chapter details S-Boun-TI, which was tested by implementing a prototype. Over one million posts that correspond to 11 post sets were gathered for testing. The post sets were retrieved during various events such as "the 2016 US Election debates", "the death of Carrie Fisher", and "the Dakota Access pipeline" demonstrations. The identified topics are published in Fuseki [42, 111]. Various SPARQL queries were run against the topics to explore the results.

To the author's knowledge, this is the first approach proposed for identifying semantic topics for a collection of microblog posts, although approaches have been proposed [62, 63] for semantic representation of single posts. The peoposed approach outputs topics that are represented in the Semantic Web. In this context, the topics of this approach are also referred to as "semantic topics". Section 6.1 introduces the approach. Section 6.2 explains the prototype details. Section 6.3 provides details of the results and an exploration of the topics.

6.1. S-Boun-TI: Structured Topic Identification

The main input of the approach is a set of microblog posts. The output of the approach is a set of topics represented with $Topic_{\mathcal{O}}$ expressions. Figure 6.3 gives an overview of the approach.

Figure 6.4 gives the algorithm for identifying topics. The details associated with these operations are described in the remainder of this section. They include identifying candidate

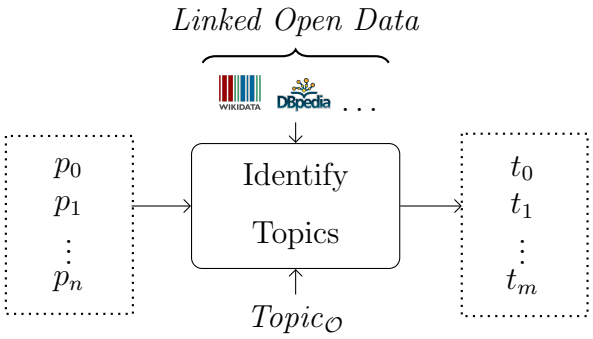


Figure 6.3. Overview of identifying semantic topics from a set of microblogs. Entities within microblog posts (p_i) are linked to semantic entities in *Linked Open Data* or to "topico:TemporalExpression" individuals, which are processed to yield semantic topics (t_j).

topic elements, improving element extraction by revisiting elements and spots with collective information, relating the elements, identifying the types of elements, and forming topics according to the relations and the types of the elements.

6.1.1. Candidate element extraction

To identify topics, elements are identified in posts. Candidate elements will be the topic elements in the next steps. The candidate elements are linked entities and the temporal expressions defined in $Topic_{\mathcal{O}}$ which are an instance of "topico:TemporalExpression". Common entity linking services such as TagMe and OpenCalais [112] can be used in this phase. Entity linking matches spots in the text and the associated web resources such as Wikipedia articles or DBpedia resources.

In the candidate element extraction phase, the linked entities are the Linked Data resources. If the entity linker does not return Linked Data resources for spots, the returned resources should be analyzed and their Linked Data resources should be identified. The purpose of this is to obtain topic elements that can be identified by other agents and further processed. For instance, the query that is introduced in the introduction of this chapter "Show me the rock music artists that people have recently been talking about along with the locations of their concerts" can be answered if the topic elements are identified as Linked Data resources. In this phase, for example, if the Linked Data resource is considered as DBpedia, the spot

1: Input: P ▷ post set
2: Output: St ▷ topic set
3: $spots, elements, types \leftarrow []$
4: $le \leftarrow []$ ▷ linked entities
5: G, G' : **graph**
6: $St \leftarrow \{\}$ ▷ semantic topics
7: ▷ identify candidate elements
8: **for each** p **in** P **do**
9: $elements[p] \leftarrow$ entities$(p) \cup$
10: mentions$(p) \cup$
11: temporalExpressions(p)
12: $spots[p] \leftarrow$ unlinkedSpots(p)
13: **end for**
14: ▷ revisit elements and spots with collective information
15: **for each** p **in** P **do**
16: $le[p] =$ reLink$(elements[p], elements)$
17: $le[p] =$ linkSpots$(spots[p], elements, spots)$
18: **end for**
19: ▷ Identify and create topics
20: $G =$ relate(le)
21: $G' =$ prune(G, τ_e)
22: **for each** v **in** G' **do**
23: $types[v] =$ getType(v, P, τ_{loc})
24: **end for**
25: $gt =$ extractTopics(G')
26: **for each** *topic* **in** gt **do**
27: St.insert(sem-topic(*topic*,*types*))
28: **end for**
29: **return** St

Figure 6.4. Semantic topic extraction algorithm that extracts topics from a given microblog post set

Hillary Clinton gets linked to "http://dbpedia.org/resource/Hillary_Clinton" in DBpedia (see Section 3.5).

Once the candidate elements are identified, their types can be determined by using Linked Data. One type of candidate element is a linked entity, which can be an agent or a location. If an element is not of this type, it will be referred to as another type of entity. A candidate element may also be a temporal expression.

While the entities returned by entity linkers are referred to as linked entities, in this thesis the temporal expressions which are individuals of "topico:TemporalExpression" are actually Semantic Web resources. Therefore, they can be considered to be entities. If they are linked with a spot in a post set, the spot becomes a linked spot and the entity can be referred to as a linked entity in this context. If this is considered, the terms candidate element and linked entity can be used interchangeably, because all linked entities become candidate elements.

If an element is of one of the following types: persons, locations or temporal terms, it is related to a topic individual that is created with one of the following properties, as appropriate: "topico:hasAgent", "topico:hasLocation", or "topico:hasTemporalExpression" of "topico:Topic". If an element is not one of these types it is related to the topic with the "topico:isAbout" property of "topico:Topic".

Some spots may not be linked due to low confidence scores of the entity linkers. These spots are considered to be unlinked spots. These spots are processed once again to see whether they can be linked to an entity. Alternatively, some linked entities may be linked to a wrong entity. These linkings are also improved. The details of the improvement of links and the linking of unlinked spots are given in Section 6.1.2. The outputs of this step are candidate elements and unlinked spots.

6.1.1.1. Agent identification. Of all the elements in microblog posts, "agents" are one of the most frequently referred to types. Therefore, they should be expressed in topics. $Topic_{\mathcal{O}}$ has special definitions for expressing agents, but agents need to be identified in order to be expressed as agents. To identify agents, their usage in microblogs should be inspected. In

microblogs, agents are referred to in two ways: with the name of the agent in post texts or with the handle of a user of the microblog system (i.e. a mention on Twitter). Spots that are user handles should be linked to entities in Linked Data whenever possible. For example, in Twitter, the @BarackObama mention is a user handle for the 44^{th} US President "Barack Obama". There is an entry in DBpedia with the identifier: "http://dbpedia.org/resource /Barack_Obama" for him. The spot @BarackObama should be linked to its Semantic Web identifier; in this case the DBpedia identifier.

The special definition for agents in *Topic*$_{\mathcal{O}}$ is the property "topico:hasAgent" and its sub-properties. The range of "topico:hasAgent" is "foaf:Agent" which has three subclasses: "foaf:Person", "foaf:Organization" and "foaf:Group". If the value of the "rdf:type" property is "foaf:Person", it can be assumed that the entity represents a person. Naturally, this depends on the quality of entity descriptions. Some resources are not described extensively or correctly. However, with time this is improving.

6.1.1.2. Location identification. Another type of element often related to microblog topics is locations. Identifying locations of topics can be done in two ways. One way is the posting source. For instance, microblog platform Twitter's mobile client and other third party mobile clients that work with Twitter can provide location information of the poster if the poster permits. Or, the poster may provide location information in his/her profile. Sometimes, users check in using applications such as Foursquare [113] that create a post about where the user is and what he/she is doing. These sources may give location information about the topics. If such information is identified, it can be expressed in the Semantic Web. *Topic*$_{\mathcal{O}}$ can be used to express this kind of information.

The problem, however, is that users often do not permit the longitude and latitude information to be attached to their posts, and not all users set the location information on their profiles. Even if this information is provided, the topic that is discussed may not be related to the poster's declared or attached location. For these reasons, only post texts are considered for location identification.

While some location referrals may be found in Linked Data, others are not found. For instance, "study hall" is a phrase that Boğaziçi University students often use to refer to a

place where they study together. This place is not expressed in Linked Data. The “study hall” phrase redirects to a resource in DBpedia that is not related to the study hall in Boğaziçi University. The locations that are expressed in Linked Data are well known locations such as “Vodafone Arena” and “İstanbul”. Identification of a location in posts that does not exist in Linked Data requires the creation of a web resource for that location so that it can be referred to in the Semantic Web. Then, that location should be related to other information that is identified. For instance, for the example “study hall”, a resource should be created. If “Boğaziçi University” is identified as related to the identified “study hall” using methods such as checking the frequency of co-occurrence in the same post, a relation should be established between the Semantic Web resource that is created and the “Boğaziçi University” which is expressed in Linked Data such as “dbr:Boğaziçi_University”. Relations to other identified entities should be expressed too. To identify if the spot is a location or not, NLP approaches can be used. For example, a phrase having the term “hall” may indicate a location. This task requires further investigation. Even for locations that are expressed in Linked Data, problems exist such as identifying if that location is used as a location in posts. Thus, identification of locations absent from Linked Data is planned for future improvements.

In this approach, in order to identify locations, the type of identified element is checked. For example, the value of “rdf:type” of a resource of DBpedia is inspected to examine if it is a location related value. Some examples of location related types are “geo:SpatialThing”, “geonames:Feature”, and “schema:Place”. Upon inspection, it has been observed that the presence of a location indicating type is not sufficient to determine that an entity is actually of a location type, since the entities that have location types also have other types. For example, for entity “http://dbpedia.org/resource/Stanford_University” linked to the spot “Stanford University” has many “rdf:type” properties including “geo:SpatialThing”, “dbo:Agent”, “dbo:Organisation”, and “dbo:University”. The type of this entity depends upon the context in which it was used in the post.

For this reason, the usage of the spot should be inspected to determine if it is a location or not. In the post “Stanford University’s Central Energy Facility by ZGF Architects is a Top Ten Green Project award winner. http://bit.ly/1NpoAXe” the type of the entity ”Stanford” is not a location, whereas in the post “I’m at Stanford Medical Practice in Brighton, Brighton and Hove” it is a location. Therefore, to identify if a candidate element is a location, usage

of their spots is considered. Elements with location related properties such as an instance of "geo:SpatialThing" class are considered locations only if their corresponding spots occur after the prepositions "in", "on", and "at" and if they sufficiently occur in that pattern ($\frac{preposition\#(entity,posts)}{\#(posts)} > \tau_{loc}$). These prepositions are obtained from observation. While this operation may not be sufficient to identify all locations, in the results obtained it has been observed that the identified locations were correct.

Location identification is generally a difficult task. It is even more difficult in microblogs due to the way micro posts are constructed. The results and challenges related to locations are further discussed in Section 6.3.2 and Chapter 8.

6.1.2. Candidate element improvement

The candidate elements obtained by processing individual posts are entities linked to spots within posts. The entities are Semantic Web resources. In different posts, the same spot may be linked to different entities due to the context of a single post text. This approach examines the candidate elements and the corresponding spots, and considers the correct entity to be the one that is most often linked to that spot. All spots that are linked to different entities are then linked to the correct entity. This is referred to as relinking.

The inspection of numerous entity linking tests revealed that, in most cases, there is a dominant entity linking for spots. It is not the case that relinking occurs among links with high frequency. Also, there may be some spots that remain unlinked. In such a case, again, the most commonly linked entity is considered the correct link, and the unlinked spot is linked to that entity. Thus, in this approach, in addition to entity links obtained from single posts, the collective contributions are used to correct some linked entities and to link unlinked spots.

A weight is given to each entity indicating its presence in a post collection. This is simply the ratio of the number of posts it corresponds to among all the posts. When entities are relinked or new entity links occur, the previous weights must be adjusted to reflect the increase in weight. At the end of this process the determination of candidate elements is finalized. A spot may exist in only one candidate element. Different spots may be linked to

the same entities. All unlinked spots are eliminated.

6.1.3. Relating elements

In this approach, related elements are considered topics. There are two tasks for obtaining semantic topics: identifying related elements and expressing them as semantic topics. The first step is relating the candidate elements.

The relationships among the candidate elements are represented with a weighted undirected co-occurrence graph. The vertices correspond to the entities and the edges connect entities that are linked to co-occurring spots. The weight of vertices corresponds to how many posts it occurs in, which is normalized by the number of posts. Let $G = (V, E)$ be the entity co-occurrence graph, $V = \{v_1, v_2, ..., v_n\}$ be the set of vertices. Let $E = \{e_1, e_2, ..., e_m\}$ be the set of edges, where each edge is $(v_i, v_j) \in V$. Let $w\colon E \to [0, 1]$ be a function that returns the weight of an edge. The weight of an edge shows the strength of the relationship between two elements. In order to represent topics relevant to many people, the elements that occur rarely are removed. The edges in G where $w(e) < \tau_e$ are considered weak and removed, where τ_e is a threshold manually set before applying the approach. The vertices that get disconnected due to edge removal are also removed. In sections 6.2.2 and 6.3.3 the values of thresholds and the use of thresholds are discussed further.

The equations 6.1 and 6.2 are applied to $G = (V, E)$ to obtain $G' = (V', E')$

$$E' = \{e | e \in E \wedge w(e) > \tau_e\} \tag{6.1}$$

$$V' = \{v | \exists e, \exists x [e \in E' \wedge (e = (v, x) \vee e = (x, v))]\} \tag{6.2}$$

The output of this process is a co-occurrence graph of non-weak elements.

6.1.4. Identifying topics

The final step of producing semantic topics consists of identifying topics within G' and instantiating semantic topics. A topic is a set of related elements. Each vertex in this graph is a candidate topic element. A set of subgraphs of G' that consist of strongly related elements

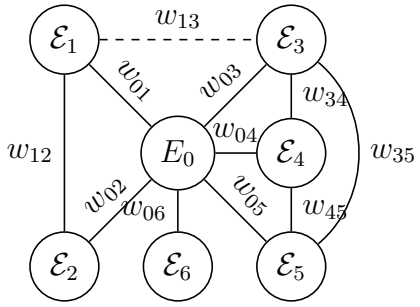


Figure 6.5. Related entities are represented with a graph, where the relation is weighted according to co-occurrence frequency. Edges with weight $< \tau_e$ are considered weak and discarded. $w_{13} < \tau_e$. In this case, when maximal clique is considered three topics emerge: $\{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2\}$, $\{\mathcal{E}_0, \mathcal{E}_6\}$, and $\{\mathcal{E}_0, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5\}$.

is computed. In the prototype implementation, the maximal cliques algorithm is employed for this task (see Section 6.2.3)

The output of this process is the set of topics T. It is the subset of the power set of V': $T \subset \mathcal{P}(V')$. Figure 6.5 shows an example graph and its topics. See implementation details in Section 6.2.2. Each $t \in T$ is mapped to an instance of "topico:Topic".

6.2. Prototype

A prototype of the proposed approach is implemented to examine its utility. The implementation gets tweets from Twitter API, processes them, and outputs topics expressed as RDF. Figure 6.6 shows a general overview of the system. During processing, TagMe is used as an entity linker tool. In operations, generally PHP: Hypertext Preprocessor (PHP) Command Line Interface scripting language is utilized. It supports JSON objects, and it has rich array and string functions. It also natively supports UTF-8 encoding, as the microblog posts have special characters that need to be handled. RDF triples of topics extracted from posts have been stored in Fuseki.

To create topics from graph G', a strict method was used to create a baseline, which is the maximal cliques algorithm. The output of the maximal cliques algorithm is processed to form topics.

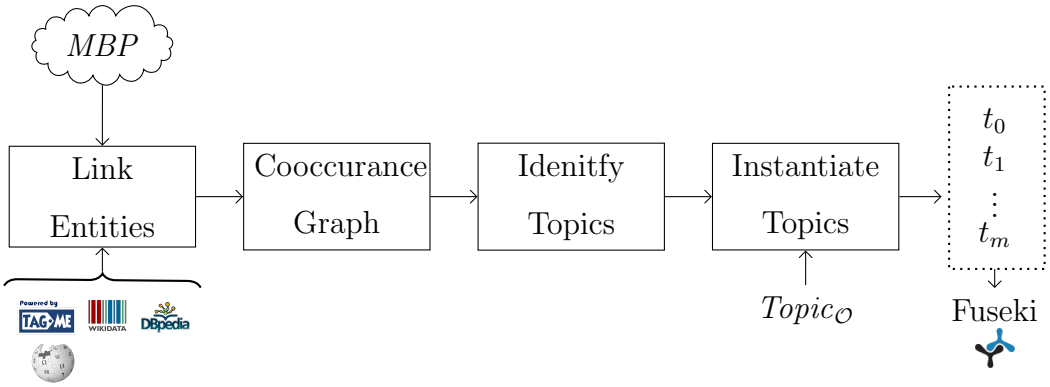


Figure 6.6. Overview of topic extraction from a set of microblog posts. $Topic_{\mathcal{O}}$ is the proposed ontology for representing topics.

The algorithm of all processes is given in Figure 6.4. In practice, however, there are small differences based on implementation details. For example, for each entity, the types are identified by calling "getType" function. But, in the implementation, the types of entities have been identified in about fifty-by-fifty chunks by calling DBpedia SPARQL endpoint. The chunk size is set according to the maximum GET method URL length. Furthermore, several tasks are accelerated by caching contents of URLs of the API calls that need to be done in tasks, since for each tweet user handles and other entities should be identified, and for each post set the types of entities are computed. Caching reduced the number of API calls to Wikidata, DBpedia and TagMe. For example, if a Semantic Web resource of a user handle has already been fetched before, it is not requested again. Similarly, if a tweet text is retweeted several times, the request to TagMe for identifying entities are always the same. Thus, the request is not made, and the response is instead retrieved from the cache.

The implemented steps are as follows: mention linking, temporal term identification and linking, pre-processing posts before entity linking, processing candidate elements with collective information, location identification, extracting the co-occurrence graph, preparing the graph for maximal cliques identification, post-processing the output of the maximal cliques identification step, and expressing the topics in RDF. Other tasks such as the maximal cliques and entity linking are done with the help of external API calls and tools.

6.2.1. Candidate element identification

Candidate elements of topics are linked entities. In this implementation, entities are linked with three methods. The first method identifies user handles (mentions). The details of this method are given in Section 6.2.1.1. The second method identifies temporal terms. The details of this method are given in Section 6.2.1.3. The third method utilizes the entity linker software TagMe.

TagMe relates part of texts to Wikipedia resource identifiers. Wikipedia resource identifiers in the English version start with "http://en.wikipedia.org/wiki/". DBpedia resource identifiers start with "http://dbpedia.org/resource/". DBpedia and Wikipedia share the same resource identifiers that come after these strings. For instance "http://dbpedia.org/resource/Barack_Obama" and "http://en.wikipedia.org/wiki/Barack_Obama" refer to the same person. This relation between Wikipedia and DBpedia is used to refer to the corresponding Semantic Web resource once an entity is linked by TagMe.

Entity link confidence threshold τ_ρ and entity spot confidence threshold τ_p are applied to the entities returned by the TagMe API. For a post, an entity link is accepted if these confidence values are above the thresholds. If the thresholds are set higher, a smaller number of spots are linked to entities. TagMe suggests a threshold between 0.1 and 0.3 for τ_ρ for better accuracy. While working on the post sets, it has been observed that the links of $\rho > 0.1$ are mostly correct, but incorrect entities still exist. While this threshold increases the correct entity linking ratio over all entity linking, the number of linked entities decreases. To lower the possibility of incorrect entities, the value is set as $\tau_\rho = 0.15$. To account for the highest number of spots, the threshold τ_p was set to $\tau_p = 0.4$. While the threshold τ_p increases, correct entities are also filtered out, leaving their spots unlinked. While the threshold decreases, more spots are taken into account. While setting the thresholds, by manual observation, a balance is sought between the number of entities that are considered linked and the accuracy of these entities.

It has been observed that entity names returned by TagMe may refer to a year, such as in "United States presidential election debates, 2012". Year referrals are important for entity identification when the source medium (such as Twitter) is highly temporal. In this

example, microblog users may actually be talking about the 2012 debates, the entity linker may decide to return this entity due to its popularity, or it may be the only candidate for the spot. It is assumed that if users are talking about past events they refer to the year, since in microblogs the content is usually about recent context and anything that is not about the current context is expressed using absolute expressions such as "in 2012 ...". In these cases it is assumed that if the post text of the entity references the same year, the entity is considered correct. If the post text does not reference a year while the entity name does, or if the post text references a year that does not match the year reference in the entity name, the entity link is removed.

6.2.1.1. Person identification. As agents of topics, "foaf:Person" type identification was the main focus in the implementation, because DBpedia often provides reliable information for individuals having "rdf:type" "foaf:Person". It has been observed that there is no "foaf:Group" type in DBpedia. Lack of organization of related statements has also been observed.

Identification of entities of "foaf:Organization" type is non-trivial. At the time this thesis was written, DBpedia had resources of type "foaf:Organization" [114], which are mostly scholarly-related entities like schools, libraries, and universities. However, there are many more resources which could be expected for type "foaf:Organization", such as "dbr:Google" and "dbr:Turkey". These entities do not have type "foaf:Organization". But, "dbr:Google" has a related type which is "dbo:Organization" and "dbr:Turkey" has location-related types. Even if "dbo:Organization" is considered, the problem of organization identification is still not solved because "dbr:Turkey" does not have this type. It may be the case that "Turkey" acts as an agent. This can be observed in posts. For example, in the post text "Turkey revokes passport of @okcthunder's star while in Romanian airport." "Turkey" is referred to as an agent. DBpedia does not account for this possibility. Expressing "Turkey" as an agent - specifically as an organization - is necessary, but could not be found as an option.

Another problem is that it is not clearly stated how DBpedia approaches "foaf:Organization" type. DBpedia often expresses organizations using "dbo:Organization". It requires further investigation to determine if having this type implies having "foaf:Organization". Manual observation of some of the "dbo:Organization" typed entities shows that these orga-

nizations can also be expressed using “foaf:Organization” such as “dbr:Google”. However, in DBpedia ontology, “dbo:Organization” is not listed as an equivalent class to “foaf:Organization”. This is interesting because “dbo:Person” is considered equivalent to “foaf:Person”.

Considering that Linked Data also evolves, uncertainty exists about organization types. Not a single entity exists of type “foaf:Group”. Identifying entities of types “foaf:Group” and “foaf:Organization” requires further investigation.

To identify persons that are not Twitter user mentions, the entity linker’s output is used. In this implementation, a SPARQL query is issued to DBpedia endpoint to get the type of entity. An entity is considered to be a person if one of the types of the entity (“rdf:type”) is “foaf:Person”.

Mentions are identified if they have a corresponding DBpedia resource. DBpedia does not provide the Twitter handles of well-known persons, but Wikidata does provide this information. Therefore, Wikidata is queried to identify resources for well known persons from microblog user mentions. For example, Twitter user @HillaryClinton and Wikipedia page “http://en.wikipedia.org/wiki/Hillary_Clinton” are given as properties of the resource for “Hillary Clinton” in Wikidata. Once the Wikidata resource is identified, the Wikipedia page property value is retrieved. This value is the Wikipedia page URL. This URL is converted to a DBpedia resource URL.

It has also been observed that resolving Wikipedia pages for Twitter user mentions provides context information to TagMe. If a page is found for a user mention, the user mention text is replaced with the Wikipedia page identifier, which is the title of the page with the spaces replaced with underscores (_). This way, in the short text, some context is provided which is used in entity linking. Thus, in implementation, first, the mentions are identified, then the entities are linked and lastly, the type of entity is checked.

6.2.1.2. Location identification. Location related types that are considered for type checking are “schema:Place”, “dbo:PopulatedPlace”, “dbo:Place”, “dbo:Location”, “dbo:Settlement”, “geo:SpatialThing”, and “umbel:PopulatedPlace”. The location identification operations are applied as given in Section 6.1.1.2.

6.2.1.3. Temporal expression identification. Terms that people often use to indicate the time of the topic, such as “Now”, “Tomorrow”, and “Tonight”, are identified and linked to the corresponding Semantic Web resource that are defined in $Topic_{\mathcal{O}}$. A look up method is used to identify these temporal expressions. For example, if a tweet text has one of the spots “tdy” or “today”, that spot is linked to “topico:Today”. If it has “yesterday” or “ystrdy”, the spot is linked to “topico:Yesterday” and if it has “saturday”, the spot is linked to “time:Saturday”. Forty-two of these rules, including seasons and month names, have been defined. These are listed in Table B.1.

6.2.2. Relating elements

From each post, entities are extracted that become candidate elements for topics. Co-occurrences of each element are computed and the graph G is formed. From graph G, graph G' is computed as shown in equations 6.1 and 6.2. Edge removal threshold τ_e is used to filter out less frequent edges to obtain a co-occurrence graph of elements that most people talk about. τ_e is set to $\tau_e = 0.001$. If this value is increased, a lower number of edges and vertices are considered for maximal cliques algorithm. This results in fewer topics. When setting the threshold, the number of co-occurrences of two elements in one post is considered. Maximum weight of an edge can be 1.0 if all the posts have the two elements of the edge. In this implementation, 0.001 corresponds to five posts for a set of five thousand posts. It is assumed that this is a sufficient duplication amount for an edge to be kept and its existence to be confirmed by multiple users. Section 6.3.3 provides information about the weights of edges and the suitability of this value.

6.2.3. Topic identification

Once the relatedness graph (G') is composed, the next step is to form the semantic topics. From the graph, maximal cliques are extracted. Once the cliques are obtained they are processed. The maximal cliques algorithm typically yields numerous edges (cliques of size two) and few k-cliques, $k > 2$. In order to gain insight into the usefulness of edges, the occurrences of the vertices (elements) in the data set have been examined. This revealed that one of the elements occurs far fewer times than the other ($freq(v_0) > 500$, $freq(v_1) < 50$) - for a post set of 5,760 posts - where $freq(v)$ returns the frequency of a vertex in the post set.

Furthermore, it has been observed that frequent element v_0 is to be in k-cliques, $k > 2$.

From post sets, 4,555 edges were obtained. The frequency ratio of higher frequency element over lower frequency element of each clique was computed. It was found that 3,186 (69%) of cliques have the ratio over ten, 2,621 (57%) of cliques have the ratio over 20, 2,268 (49%) of cliques have the ratio over 30, 1,925 (42%) of cliques have the ratio over 40. 1,683 (36%) of cliques have the ratio over 50. These findings show that there is a gap between the frequencies of elements of edges. An edge is considered as a topic if both elements are frequently used. An edge is removed if the frequency of one of the elements is below threshold τ_{two}, a threshold that is expected to be set before application of the approach. This threshold has been set to 0.01, which is high according to most of the edges extracted from tweet sets. Section 6.3.3 provides typical weights of edges. According to these observations, the edges that are common among posts are selected.

It has been observed that some cliques are similar due to having many of the same elements. These cliques are separated by the maximal cliques algorithm because of the edge removal threshold that is applied before running the algorithm. Examples of such cliques are {Hillary_Clinton, Donald_Trump, 2016, Answer, Muslim} and {Hillary_Clinton, Donald_Trump, 2016, Question, Muslim} These cliques are similar, but each has distinct elements which are "Answer" and "Question". They are distinct because some of the relationship weights between these elements and the other clique is below τ_e. Two cliques are merged if they are similar and if the elements of one clique are related to all the elements in the other clique and vice versa. But this time, the threshold weight of the edge between two vertices (elements) is $\tau_{e_{min}}$, which is lower than τ_e. $\tau_{e_{min}}$ is a threshold which is manually set before running the approach. In the implementation, $\tau_{e_{min}} = 0.0005$. Cliques are considered to be similar if their Jaccard coefficient is above threshold τ_c. τ_c is another threshold which is manually set before running the approach. In the implementation $\tau_c = 0.8$, which is considered similar enough.

Let T be the set of cliques obtained after applying the maximal cliques algorithm. T is subset of the power set of V'. t_i and t_j are removed and $t_i \cup t_j$ is added to T if $jaccard(t_i, t_j) > \tau_c$ and for all $t_i \in T$, $t_j \in T$, $e_x \in t_i$, $e_y \in t_j$, $w((e_x, e_y)) > \tau_{e_{min}}$.

6.2.4. Semantic topic instantiation

Creating topics is straightforward. Once cliques are obtained and processed, the final form of the cliques are output as topic instances. For each clique, an instance of "topico:Topic" class is created. The relationship type between each element and the topic instance is decided according to the element type. For example, if the entity is a person, "topico:hasPerson" is selected, if the entity is a temporal term "topico:hasTemporalTerm" is selected, and if the entity is a location "topico:hasLocation" is selected. For others, "topico:isAbout" is selected. Meta information such as the topic observation interval is added by creating a "time:Interval" instance and relationship. "topico:hasObservationInterval" and topic creation time is added with data property "topic:topicCreatedAt".

6.3. Experiments and results

Since this approach is the first to produce topics of microblog post sets to be expressed in the Semantic Web, a gold-standard dataset does not exist for comparison. A prototype was implemented and the results of the prototype were evaluated. This section explains the properties of the datasets and evaluates the approach in four sections: The quality and the quantity of linked entities, the obtained co-occurrence graphs and cliques, and processing of the resulting topics.

6.3.1. Post set characteristics

In order to work with a sufficient number of tweets that were expected to persist over a long time and serve as an interesting context, tweets related to the 2016 US presidential election were collected. The volume of this data was particularly high during the presidential debates and the vice presidential debate, which were processed in two minute intervals. The majority of the data is related to politics and the election (1,032,804/1,071,757). Other datasets were concerned with concerts, the Dakota Access Pipeline protests, Carrie Fisher, Brangelina, Toni Braxton, and the inauguration. Table 6.1 shows the keywords that the post sets queried using Twitter Streaming API. In most of the sets, unique user ratio is higher than 0.70 which suggests that the topics are obtained from posts by different users. This is a result that is desired since S-Boun-TI extracts topics of a collection from posts from multiple

Table 6.1. Characteristics of post sets collected from Twitter.

Set (#)	**Query (keywords)**	**Start time (UTC)**	**Duration (min.)**	**Posts (cnt)**	**Distinct Poster**	
					(cnt)	**%**
1	*PD kw*	2016-09-27T 01:00:00Z	90	259,200	206,827	79
2	*PD kw*	2016-10-10T 01:00:00Z	90	259,203	187,049	72
3	*PD kw*	2016-10-20T 01:00:00Z	90	258,227	181,436	70
4	*VPD kw*	2016-10-05T 01:00:00Z	90	256,174	135,565	52
5	#Brangelina	2016-09-20T 23:38:38Z	503	6,000	4,777	79
6	Carrie Fisher	2016-12-28T 13:59:50Z	15	7,932	6,753	85
7	concert	2016-12-02T 19:00:00Z	60	5,326	4,743	89
8	north dakota	2016-12-03T 06:59:48Z	14	7,466	6,231	83
9	Toni Braxton	2017-01-08T 07:08:56Z	765	5,948	4,506	75
10	#inauguration Trump @realDonaldTrump	2017-01-21T 20:41:44Z	6	5,809	5,425	93
11	*(no keyword)*	2016-12-02T 20:29:53Z	8	5,472	5,365	98

users; this result shows that the topics are extracted from posts mostly from different users.

6.3.2. Entity linking

In order to gain insight into the identified topics, the entities within topics were examined. Table 6.2 shows the percentage of posts whose elements (mentions, temporal expressions, persons, locations, and other) were linked to entities. This showed that posts include mentions that can be linked to web resources. Only three datasets have a low mention linking ratio (< 0.07) which are the public stream (set 11), #Brangelina (set 5), and Toni Braxton (set 9). The public stream is expected to have low mention links since the stream is not in the context of any specific subject, but a sample of all English tweets including mentions not only of well-known Twitter users but of all possible Twitter users. In Brangelina and Toni Braxton sets, people talked heavily about the divorce of Brad Pitt and Angelina Jolie and the death of Toni Braxton by referring to them by their names, but not using Twitter user mentions. Wikidata does not give Twitter usernames for Brad Pitt and Angelina Jolie. Twitter is also searched, and it is found that Angelina Jolie and Brad Pitt do not have validated Twitter accounts. Toni Braxton, on the other hand, has a Twitter account, but in the tweet set, only 24 out of 5,948 tweets mention her Twitter username. In the 2016 US presidential election debate sets, it is observed that 79% of the posts have at least one element and 36% of the posts have at least one relationship (two elements). For example, in the debate datasets, the following mention links were found: @HillaryClinton to "dbr:Hillary_Clinton", @VICE to "dbr:Vice_ (magazine)", @aplusk to "dbr:Ashton_Kutcher", and @DonaldJTrumpJr to "dbr:Donald_Trump_Jr".

It can be seen that in some cases, mention linking improved entity linking. For example, entities of the tweet "@GeraldoRivera @realDonaldTrump @ApprenticeNBC @HillaryClinton Clinton wor the wrong color tonite. ... makes her look older! (Like 100 yo)" were identified as "dbr:Bill_Clinton" and "dbr:WOR_(AM)". Both were incorrectly identified in the initial phase.

Once the @GeraldoRivera, @realDonaldTrump, @ApprenticeNBC, @HillaryClinton are replaced with "Geraldo_Rivera", "Donald_Trump", "Apprentice", "Hillary_ Clinton" respectively, the text is transformed to "Geraldo_Rivera Donald_Trump Apprentice" "Hillary_Clin-

Table 6.2. The types of entities in the post sets

Set	Mention	Time	Person	Location	Other
1	29	22	69	3	85
2	43	18	88	5	74
3	31	9	70	8	85
4	35	10	70	6	88
5	4	13	17	4	31
6	17	17	61	4	50
7	18	26	16	15	91
8	18	11	5	77	84
9	0.5	7	31	0.1	45
10	27	14	54	17	68
11	7	12	6	5	38

ton Clinton wor the wrong color tonite. ...makes her look older! (Like 100 yo)". After entity linking is applied to this text, in addition to the entities that were already identified by mention linking, "dbr:WOR_(AM)" was returned, which is incorrect. Only one incorrect entity is found after mentions are replaced. An incorrect link from the spot "Clinton" to "dbr:Bill_Clinton" was eliminated.

Spots exist that should have been linked but could not be linked due to the threshold applied to the output of the entity linking tool. An example of such a spot is "hillary". In some cases, the spot "hillary" could be linked to "dbr:Hillary_Clinton" and in others not. The correct linking is to "dbr:Hillary_Clinton". For example, in one of the experiments, ([40-42)nd minutes of the second presidential debate of 2016), the spot "hillary" could not be linked in 17% of the posts. Likewise, "trump" could not be linked in 6% of the posts. These spots are linked with correct entities using collective information of linkings as given in Section 6.1.2.

Linked entities have confidence values above thresholds τ_ρ and τ_p. However, some of these entities are linked incorrectly for a variety of reasons. The spots of these entities are re-linked as explained in Section 6.1.2. Table C.1 provides some of the spots of these

entities and their linked entities before and after re-linking. An example of an incorrect entity is "dbr:George_III_of_ the_United_Kingdom" which is extracted from the tweet text "RT @sykojuicee: How did 2016 take Bowie, Prince, George Michael, Alan Rickman, Carrie Fisher and so many more but leave that f...g cheeto...". Instead of linking the spot "Prince" to "dbr:Prince_(musician)", and "George Michael" to "dbr:George_Michael" the tool linked "Prince, George Michael" as a whole to "dbr:George_III_of_the_United_Kingdom" due to the word "Prince" being frequent in the Wikipedia page of "George_III_of_the_United_Kingdom". Another example of this type of incorrect link is from the post text "The awesome life and career of 'Star Wars' icon Carrie Fisher". In this post, the spot "icon" is linked to "dbr:Icon_(computing)" which is incorrect. In this example, we would not expect the spot to be linked to any resource since Wikipedia does not include any page for the intended meaning of "icon" at the time this article was written.

One type of element is temporal expressions. Due to the importance of the temporal aspects of microblogs, special attention was paid to capturing temporal information. The temporal aspect of topics are combined with the meta information of post time along with temporal references present in posts. Time references in general were caught correctly, but challenges were encountered due to the ambiguous terms containing month names, such as "May" of Theresa May, "March" of the "women's march" held on 21 Nov 2017, the day after the inauguration of President Trump.

Among the recognized temporal expressions, relative references such as "tonight, tomorrow, now, today" were identified correctly. Years were also identified correctly. Month names, however, were identified incorrectly. For example, in the dataset about Carrie Fisher, users often refer to the phrase "May the force be with you". The word "May" is not a month name in this context. But it is recognized as a temporal term and appeared as a temporal term in three of the topics. Month names such as "April", "May" and the season name "Summer" can be used to refer to people. The season name "Fall" has the same spot with the verb fall. Thus, a more sophisticated approach is needed to handle these cases. On the other hand, all temporal expressions including month and season names were correctly identified in the tweet set about concerts. Example use cases are: "fall concert", "summer festival", "Winter Concert", "WinterFest".

Another type of element is locations. It is observed that all the locations in topics were correct. Once the tweets are investigated, it can be seen that in the debate sets people refer to locations a few times. At first this result would not be expected, but after investigating the tweets it is observed that people talk about political issues, not specific locations. On the other hand, the concert and inauguration datasets resulted in more locations. This result is expected, since in the concert dataset people talk about concerts and related places, and during the time the inauguration dataset was fetched, the Women's March was being held and people were talking about protest locations. The results show that the North Dakota dataset has the highest ratio of location-related posts. This result is expected since the posts have "North Dakota" in the text which is identified as a location.

It is observed that some location referrals are missed due to the strict rules of location identification such as thresholds and prepositions. Identifying locations requires sophisticated methods that differentiate if a user is mentioning an entity as a location, an organization or simply refers to it as a regular entity.

Overall, linking unlinked spots using collective information and user mentions is shown to increase performance in creating topics and forming new entity links. On average, in post sets [1-4], 21% of posts, and in post sets [5-11], 11% of posts provided new entity links. These results suggest that new entity links are introduced in posts in the collective information processing phase. These entities are considered when computing edge weights of the graph formation in the next steps of the approach.

6.3.3. Co-occurrence graphs and cliques

While working on S-Boun-TI, the aim was to extract topics that most people talk about. To do this, S-Boun-TI extracts weighted relationships, where weights are made up of individual posts by different users. It creates a graph of the elements and their relationships and prunes this graph by removing edges that have weights below certain thresholds ($\tau_e = 0.001$ and $\tau_{e_{min}} = 0.0005$).

Firstly, the effects of pruning on the number of vertices and edges is inspected. The number of vertices and edges before and after pruning are given in Table 6.3 for each dataset.

The table suggests that about 2% of the edges are kept. It is observed that there are too

Table 6.3. The number of vertices and edges before and after pruning entity co-occurrence graphs.

Set	**Vertices**		**Edges**	
	Before	**After**	**Before**	**After**
1	939,936	14,657	163,162	3,898
2	924,816	13,669	156,088	3,855
3	934,189	16,040	163,828	4,404
4	1,110,056	19,244	172,588	4,408
5	6,687	58	424	35
6	11,540	172	610	96
7	18,575	159	1,284	106
8	10,741	191	586	97
9	4,483	508	339	80
10	18,196	186	930	88
11	22,295	25	2,130	17

many edges of low weights and fewer edges of higher weights. Figure 6.7 shows the weights and ratios of the edge weights in various sets before pruning the graph G. For comparison, different thresholds are set. It is shown that if the thresholds are set higher, a fewer number of elements make it into the topics. This case results in more abstract topics. For example, for most of the debate post sets, if the $\tau_e = 0.1$, only the edges among the elements {Donald Trump, Hillary Clinton, Debate, Tonight} are considered. Thus, only one topic is extracted consisting of these elements. The maximum value that can be set is 1.0, which suggests that each element of an element pair should be linked to a spot in each tweet. However, this situation is unlikely in the tweet sets. For example, for values $\tau_e > 0.2$, S-Boun-TI does not result in any topic since no relationship weight conforms to this condition. If the threshold is lowered, fewer edges are removed, which results in more fine-grained topics at the end like {Donald Trump, Hillary Clinton, Debate, Middle class, Trickle-down economics, Tax, Americans}, and {Donald Trump, Bashar al-Assad, Vladimir Putin, Moscow, Now, Debate}. In this case, however, edges obtained from a smaller number of tweets contribute to topics, which means that a smaller number of people contribute to the formation of topics. The

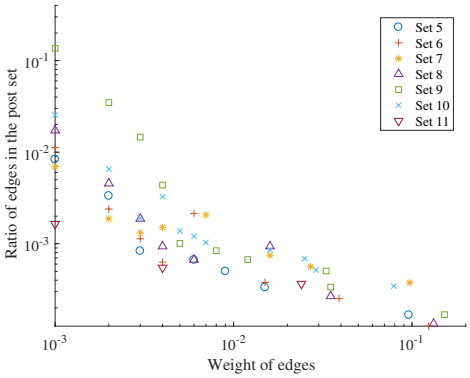


Figure 6.7. Weight of edges and the percentage of those edges in the post sets. The x and y-axis are in logarithmic scale.

minimum of this value is zero, which suggests that each relation (element pair) is accounted for even if its elements are linked to spots in only one tweet. This case results in too many topics, mostly consisting of contributions from a single person or a few people. Taking these effects into consideration, the pruning threshold is set to best extract topics that most microblog users talk about.

Table 6.4 shows the ratio of the posts that edges and vertices were extracted from before and after pruning and resulting topics. It is observed that some posts exist in which no entity is linked to a spot. Pruning did not affect a large portion of posts in which an element is linked to a spot. The ratio of posts that elements were extracted from does not drop with the number of elements after pruning. This is due to keyword-based post collection. The keywords match entities which are candidate elements of topics. This effect is not observed for the public set (set 11). Since the public set has various elements that are not related to any keyword, pruning edges resulted in fewer edges that gathered from fewer posts. The number of edges, and the ratio of posts to edges show similar behavior before and after pruning. A decrease is observed in the ratio of posts to elements and edges in the topics. This is due to elimination of some elements in 2-clique removals, and is an expected result.

The number of cliques and size of various sets have been inspected. The sizes of the cliques after running the maximal cliques algorithm are given in Figure 6.8a. Figure 6.8b shows the clique sizes after processing. The figures show that most of the edges are removed but a few of them are kept. Among the k-cliques, $k \geq 3$, the similarity check and merging

Table 6.4. The percentage of tweets that contribute to the vertices, edges and topics. The columns labeled "Before" and "Pruned" show the impact of pruning the graph. The columns labeled "Topic" show how many ended up in the topic.

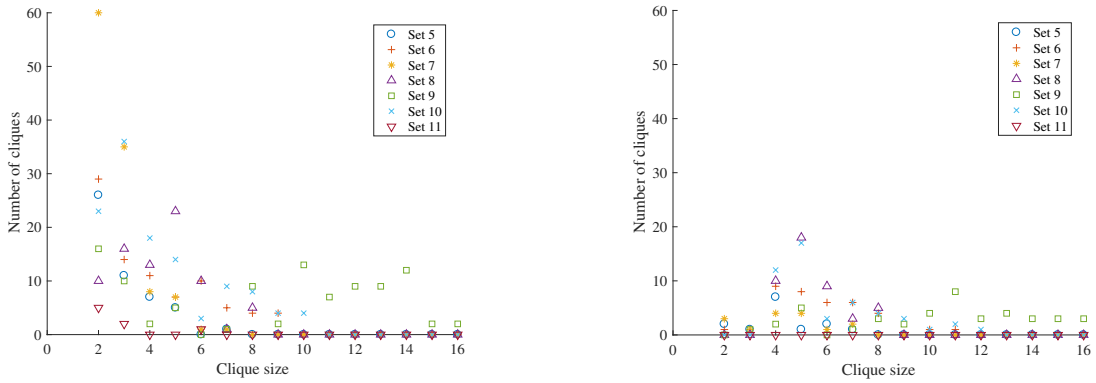
Set	**Vertices**			**Edges**		
	Before	**Pruned**	**Topic**	**Before**	**Pruned**	**Topic**
1	71	63	59	37	27	23
2	71	65	61	38	30	24
3	67	57	51	34	23	16
4	71	61	55	37	26	20
5	42	30	26	17	13	8
6	77	74	69	43	38	24
7	87	83	78	64	52	35
8	92	86	75	64	60	51
9	43	41	32	25	22	18
10	81	74	69	52	42	33
11	47	10	0	20	2	0

effect are observed. The number of cliques decreases due to the merging operation. Since similar topics are merged, this is an expected result.

From Table 6.4, it is also observed that in spite of the pruned edges and elements and the decreasing ratio of posts elements and relationships that are extracted from, the resulting topics are gathered from a representative ratio of posts.

6.3.4. About resulting topics

Table 6.5 shows the number of all topics and the number of topics that have persons, locations, and temporal expressions. It is observed that most of the topics have persons. Temporal expressions exist in the set of Carrie Fisher (set 6). Locations exist in the concert, North Dakota, inauguration, and debate datasets. Once the posts are observed, relative expressions such as "Now", "Today", "Tomorrow" and days of week are used a few times. Months of the year and seasons are referred to in the concert dataset. Applying this method



(a) (b)

Figure 6.8. Various cliques and their amounts before and after post-processing cliques. On the x-axis, the n-cliques ($n \geq 2$) and on the y-axis the number of nodes in each clique (a) after running the maximal cliques algorithm, before post-processing cliques, and (b) after post-processing cliques

to the public set resulted in zero topics.

In order to gain insight into the kinds of topics the proposed approach resulted in, an inspection tool was created. This tool displays the topics, their entities and temporal terms and allows the tweets that contributed to these topics, the post set from which they were generated, as well as those that contributed to the entities to be inspected. Figure 6.9 shows a fragment of this application that visualizes the topics and allows them to be labeled with a satisfaction level ranging from "very" to "not" as well as an optional comment. Results from the approach BOUN-TI were also juxtaposed in order to compare them. The generated topics were manually inspected by the authors of this work via this application. Manual evaluation is an intensive task requiring the evaluator to inspect the original set (5-6 thousand tweets) as well as the tweets and the DBpedia entities. While tedious, this evaluation was very useful for gaining insight regarding the resulting topics and the method. These insights are presented in this section.

In general, the resulting topics seem quite relevant, although there are some notable issues. To provide an idea for the topics, their elements will be referenced as linked entities. Linked entities are denoted by the surface form within square brackets followed by the linked entity as a url [stop and frisk]$\rightarrowtail$ ["dbr:Stop-and-frisk_in_New_York_City"].

Table 6.5. Characteristics of topics according to the number of persons, locations and temporal expressions. The values are the number of topics.

Set	**All (#)**	**Including a person (#)**	**Including a location (#)**	**Including a temporal expression (#)**
1	1,770	1,680	11	920
2	1,645	1,629	44	622
3	1,929	1,867	29	363
4	2,162	1,986	53	423
5	3	2	0	2
6	14	13	0	8
7	10	5	2	5
8	22	5	22	6
9	44	44	0	1
10	18	18	8	9
11	0	0	0	0

The resulting topics consist of entities that were related through tweets. For example, in the North Dakota dataset, numerous topics related to protests of the transatlantic pipeline system emerged, such as [President, president, US President, President of the United States, POTUS, president of the United States, US president, presidency, American President, President of United States, President of United States of America, PRESIDENT, President of the United States, Presidential, U.S. President, President of the U.S, Pres, President of the United States of America]↣ ["dbr:President_of_the_United_States"], [Obamacare]↣ ["dbr:Patient_Protection_and_ Affordable_Care_Act"], [pipeline]↣ ["dbr:Pipeline_transport"], [Bismarck]↣ ["dbr:Bismarck,_North_Dakota"], [protesters, protest, protesting, protests, Protester, Protestor]↣ ["dbr:Protest"], [standing rock]↣ ["dbr:Standing_Rock_Indian_Reservation"] from "RT OLBLightBrigade: Milwaukee is standing with #StandingRock to demand North Dakota authorities stop brutalizing water protectors". In addition to tweets dominated by the protests, other topics were identified, such as those related to a basketball game with [*Boston College*].

As expected, many people were detected, such as [Naked Cowboy]⇀ ["dbr:Naked_Cowboy"] from tweets like "WATCH: North Dakota Sen. Heidi Heitkamp boards Trump Tower elevator with the Naked Cowboy #TCOT #WakeUpAmerica #MAGA https://t.co/idDhVukDrx", [trump]⇀ ["dbr:Donald_Trump"] from "@thehill ok but why is the North Dakota senator meeting with trump over energy secretary when he owns #nodapl stock???", [Mark Ronson]⇀ ["dbr:Mark_Ronson"] from "Musicians including Mark Ronson sign open letter to Barack Obama over North Dakota pipeline protests https://t.co/4CyCbuTw9w".

References to lesser known entities and person names can be problematic. For example, Mike Pence, who was the the vice presidential candidate at the time of data collection, was often referenced as "pence" in tweets. In addition to the fact that he was not as well known at the time, the meaning of the word "pence" is also related to currency. This term was linked as [pence]⇀ ["dbr:Pound_sterling"]. Furthermore, in the same date set, the spot "kaine" was linked as [kaine]⇀ ["dbr:Kaine"], which is a fictional character. Thus, the vice presidential debate data set suffers from insufficient detection of the key persons related to the context. When tweets refer to Mike Pence or Tim Kaine, they are recognized without any trouble. Similarly, user mentions, ie. @timkaine, are linked correctly. Improved representation of semantic resources as well as entity linkers benefit all systems that rely on their results. This effect was observed when the topic identification algorithm was subsequently performed again.

There are some named entities that are not correctly identified, such as those that are not in Linked Data resources. For example, in "MSNBC reports WH has confirmed Flynn did speak to Russian ambassador re sanctions. That means Flynn lied to Pence & admin misled public." "Flynn" is not recognized and "Pence" is incorrectly recognized as "dbr:Penny". Furthermore, there are numerous matches to "Mike Flynn", with a soccer player being most prominent, so the spot "Mike Flynn" also may be incorrectly linked. Person names are difficult since they are often ambiguous. However, entity linking is continuously improving according to context. These improvements will directly benefit the proposed approach. Likewise, Linked Data resources are improving, benefiting all approaches that rely on the data they provide and compounding the improvements in results. As this thesis was being finalized, numerous beneficial updates to Wikidata were observed, which seems to be the most actively improving resource. That being said, the topic detection algorithm should further

exploit collective signals to better identify entities.

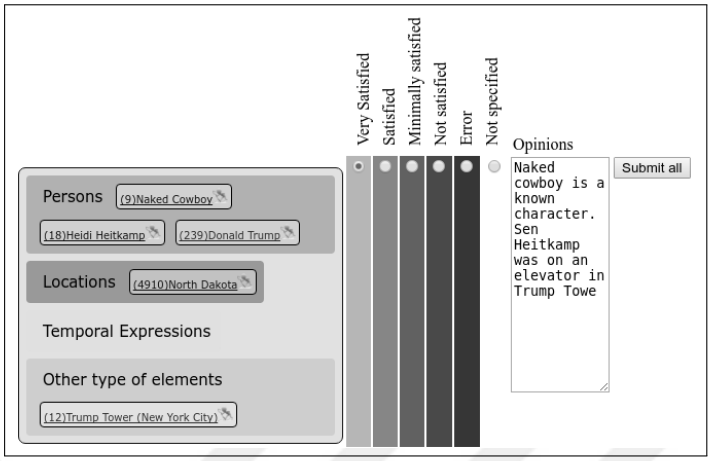


Figure 6.9. The interface to view and inspect semantic topics. This topic was caught from tweets gathered during protests in North Dakota. This topic relates to an activist known as "Naked Cowboy" from tweets of "Senator Heitkamp" as she was in an elevator with him in Trump Tower. This interface enables the inspector to view the post set, linked entities, and tweets that have contributed to a given entity.

When the processing of posts related to the inauguration of President Trump (set 10) started, it was not expected that topics that weren't directly related to the inauguration would be seen. However, topics were observed which are related to other contexts. Topics related to the "Women's March" event as well as topics about Madonna and Michael Moore, who participated in the event, were observed. Locations such as London, France and Spain were observed, as people showed their support for the event. These results suggest that the method identifies topics that are not directly related to the keywords of the post set.

6.3.5. Processing Topics

In this section, the usefulness of the topics that were gathered are presented. How the topics can be used to query information in Twitter is shown. S-Boun-TI was applied to the post sets, and the topics are expressed with $Topic_{\mathcal{O}}$.

Let us assume that we want to know who else is most often referred to in relation to Hillary Clinton. This query normally requires analysis of Twitter data. The approach expresses the topics, structured so that a SPARQL query can be issued from the structures

```
SELECT ?person (COUNT(?topic) AS ?C) WHERE {
?topic topico:hasPerson dbr:Hillary_Clinton .
?topic topico:hasPerson ?person .
FILTER (?person NOT IN (dbr:Hillary_Clinton ) )
}
GROUP BY ?person
ORDER BY DESC(?C)
```

Figure 6.10. The query for finding the people who most often appeared in the same topic with Hillary Clinton.

obtained from Twitter. Thus, if we want to learn who else is most often referred to in relation to Hillary Clinton, the SPARQL query in Figure 6.10 can be issued. This query retrieves 56 results, the first three being "dbr:Dondald_Trump", "4606"^^xsd:integer, "dbr:Bill_Clinton", "3468" ^^xsd:integer and "dbr:Tim_Kaine", "768"^^xsd:integer.

```
SELECT DISTINCT ?startTime ?endTime  WHERE {
?topic topico:observationInterval ?interval.
?interval time:hasBeginning ?begin.
?interval time:hasEnd ?end.
?begin time:inXSDDateTime ?startTime.
?end time:inXSDDateTime ?endTime.
{?topic topico:isAbout dbr:Women\'s_rights.}
UNION {?topic topico:isAbout dbr:Abortion.}
UNION {?topic topico:isAbout dbr:Women\'s_health.}
UNION {?topic topico:isAbout dbr:Violence_against_women.}
}
```

Figure 6.11. The query for determining when the topics related to "Women's Issues" were talked about, such as Women's rights, abortion, and sexual assault.

It would be interesting to explore when, if ever, an issue gained significance. This information could then be inspected to see which issues resonated with the public. To determine when topics related to women's issues emerged, a query regarding abortion, sexual assault, women's health, and violence against women could be run (see Figure 6.11).

Unfortunately, DBpedia returns “dbo:Person” as the type of “dbr:Women’s_rights”, resulting in the topic identification algorithm recognizing it as a person. Therefore, this query does not find topics about Women’s_rights. When “?topic topico:isAbout dbr:Women’s_rights” is replaced with “?topic topico:hasPerson Women’s_rights” in the query, then it matches 71 topics including topics about Women’s_rights along with their time intervals (i.e.“2016-10-05T02:24:00Z”ˆˆxsd:dateTime , “2016-10-05 T02:25:59Z”ˆˆxsd:dateTime).

Queries may indicate specific times or people. Figure 6.13 shows various issues (elements of topics) that occurred with “Hillary Clinton”, “Donald Trump” or both of them. This is a federated query with two parts to the same SPARQL endpoint of the Fuseki server. The first part selects the topmost 50 topics about Hillary Clinton or Donald Trump. The second part queries the time intervals, entities, and persons within those topics. The two queries are joined on equal entities, yielding 5,338 results (i.e. “2016-10-05T01:02:00Z”ˆˆ“xsd:dateTime” “dbr:Debate” “dbr:Hillary_Clinton”). Elements such as “dbr:Debate”, “dbr:Question”, “dbr:Answer”, and “dbr:President_of_the_United_States” were all common to Hillary Clinton and Donald Trump, thus they were removed to highlight issues that show some difference. Figure 6.12 shows the query for fetching this information.

The entities that occur with Hillary Clinton and/or Donald Trump were examined to verify the relation. The transcript of the debate revealed that “racism” was most frequently discussed in the second half of the first presidential debate and first half of third debate. The identified topics also relate to racism during the same periods. The rows for “White_people” and “Black_people” indicate this. Furthermore, the topics relate to both candidates, who were referenced either as a mention or by their first, last, or full name. Tweets were inspected, and it was observed that people from both political parties talk about this issue (pro-republicans and pro-democrats). According to the table, “Tax” is only related to Donald Trump in the [48,50)th minute of the vice presidential debate and [80,82)th minutes of the third presidential debate. It is confirmed by inspecting tweets that the issue is only related to Donald Trump in those minutes. Two tweets about “Tax” refer to Hillary Clinton in the third debate, which is below the thresholds to be considered as a relation.

Topics of S-Boun-TI can be utilized to issue queries in conjunction with knowledge in LOD. An example query is “Who are the politicians people talk about?”. This query

```
SELECT ?time ?entity ?person {
 SERVICE <http://193.140.196.97:3030/topic/sparql>
 {
  SELECT ?selectedEntity (COUNT(?topic) AS ?C)
  WHERE {
   ?topic topico:isAbout ?selectedEntity .
   ?topic topico:observationInterval ?interval .
   {?topic topico:hasPerson dbr:Hillary_Clinton}
    UNION
   {?topic topico:hasPerson dbr:Donald_Trump}
  }
  GROUP BY ?selectedEntity
  ORDER BY DESC(?C)
  LIMIT 50
 }
 SERVICE <http://193.140.196.97:3030/topic/sparql>
 {
  SELECT ?time ?entity ?person WHERE {
  ?topic topico:hasPerson ?person.
  ?topic topico:isAbout ?entity .
  ?topic topico:observationInterval ?interval .
  ?interval time:hasBeginning ?intervalStart .
  ?intervalStart time:inXSDDateTime ?time .
  FILTER(?person IN
   (dbr:Hillary_Clinton, dbr:Donald_Trump))
  }GROUP BY ?time ?entity ?person
 }
FILTER (?entity=?selectedEntity)
}
```

Figure 6.12. A query for determining when the topmost 50 topics related to Hillary Clinton and Donald Trump emerged.

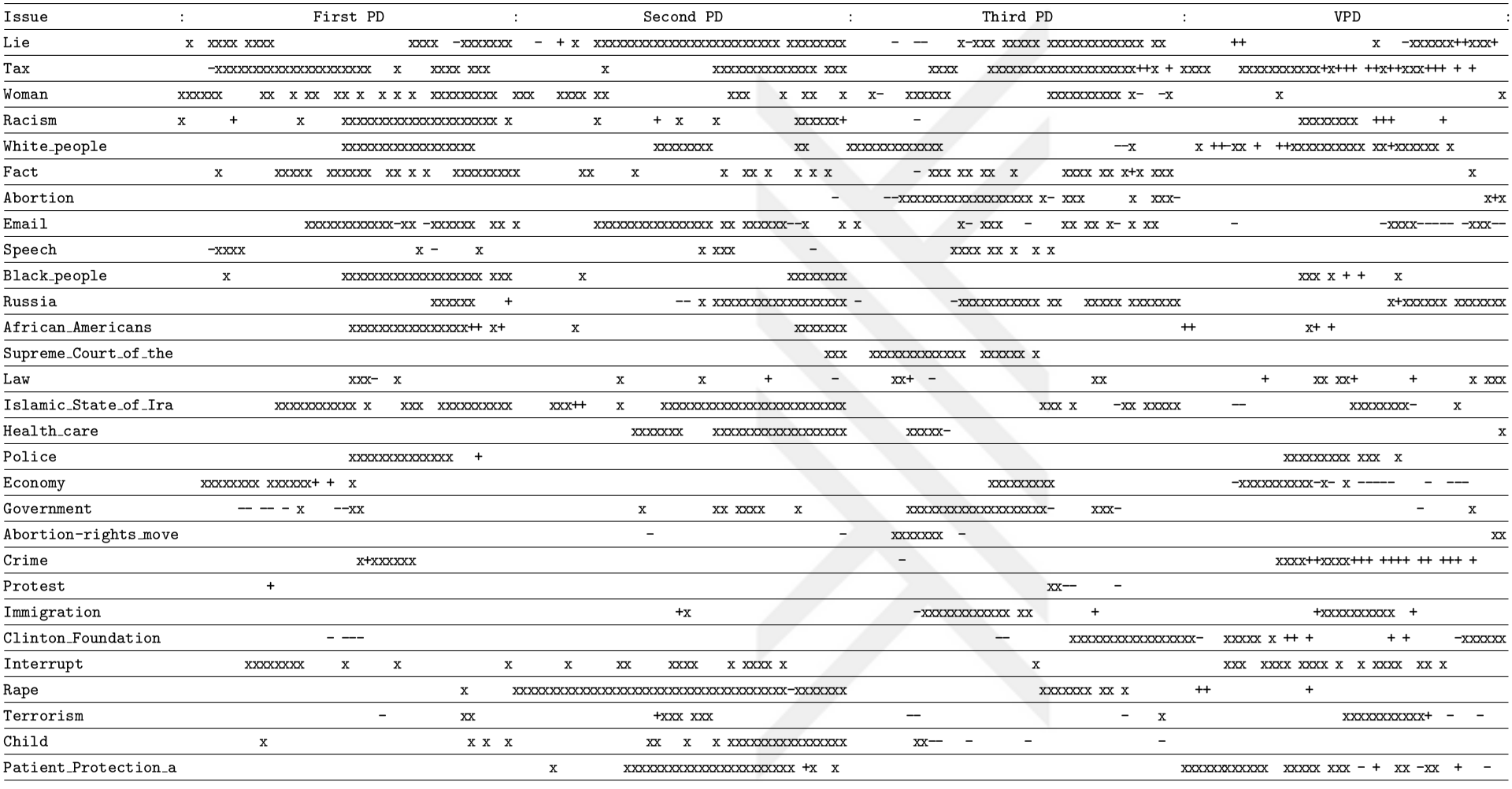


Figure 6.13. Various issues discussed during the four 2016 US presidential election debates. The symbols +, -, x indicate that the issue co-occurred with Hillary Clinton only, Donald Trump only, or both. The co-occurrence is reported for two minute intervals of each 90 minute debate.

has two parts. The information regarding which people have recently been talked about in topics comes from tweets and extracted by S-Boun-TI. However, information about their occupations are in LOD. For example, Wikidata provides the occupation of a person. Considering DBpedia provides Wikidata resource identifiers of entities, and S-Boun-TI extracts topic elements which are linked to DBpedia, three SPARQL queries can be issued to three endpoints. Firstly, all persons are extracted from S-Boun-TI topics with the first query in Figure 6.14. This query is issued to the Fuseki server SPARQL endpoint. At this point, the results can be limited to a specific time interval by setting "?topic topico:observationInterval" property.

Once the persons are obtained, DBpedia SPARQL endpoint is queried to obtain Wikidata person identifiers, as in the second query in Figure 6.14. In this query, resources in the first FILTER statement filter a set of DBpedia resource identifiers obtained from the previous query. Three dots in the FILTER statement indicate the list of DBpedia resource identifiers, separated by commas. The list is shown as dots since it is too long to fit into the provided space. The second FILTER statement filters out references to other domains except Wikidata. Once the Wikidata resources and the corresponding DBpedia resources are obtained, the third query in Figure 6.14 is issued to Wikidata endpoint hosted by DBpedia to obtain persons who are politicians. Three dots in the FILTER statement indicate the list of Wikidata resources (again, the dots are used because the list is too long). Wikidata has resource identifiers start with "Q" followed by an integer. FILTER statement filters the set of Wikidata resources obtained from the previous query. Q822955 is the resource "Politician". Once the resource identifiers of politicians are obtained from this query, the corresponding DBpedia resources can be output as an answer. The correspondences have already been obtained from the previous query. These operations were implemented and the queries were run. A list of 26 politicians was received, including "dbr:Abraham_Lincoln", "dbr:Bill_Clinton", "dbr:Colin_Powell", and "dbr:Saddam_Hussein". The full list is given in Table D.1.

To demonstrate the utilization of locations in topics, the following example is given. Let us assume that we want to query Twitter data about the recent topics that are related to hard rock music concerts, artists and locations. The query is given in Figure 6.15. This query returns one result which is "dbr:Guns_N'_Roses", "dbr:Mexico_City". If the same query is issued for Country music by replacing "dbc:Rock_music_genres" with "dbc:Country_music_genres"

```
#Query 1:
SELECT DISTINCT ?person WHERE {
?topic topico:hasPerson ?person
}

#Query 2:
SELECT ?DbPediaPerson ?wikidataPerson WHERE {
?DbPediaPerson owl:sameAs ?wikidataPerson .
FILTER (?DbPediaPerson IN
(<http://dbpedia.org/resource/Donald_Trump> ,
 <http://dbpedia.org/resource/Lester_Holt>,
 ... )
).
FILTER regex(str(?wikidataPerson), "^.*wikidata.*$")
}

#Query 3:
SELECT ?person WHERE {
?person dbpedia-owl:occupation wikidata-dbpedia:Q82955 .
FILTER (?person IN
(<http://wikidata.dbpedia.org/resource/Q22686> ,
<http://wikidata.dbpedia.org/resource/Q6294>
 ... )
) .
}
```

Figure 6.14. Query of politicians in the topics applied on three endpoints: S-Boun-TI-Fuseki, DBpedia, and DBpedia-Wikidata (in respective order).

```
SELECT ?artist ?location {
 SERVICE <http://193.140.196.97:3030/topic/sparql>
  {SELECT ?topic ?artist ?location WHERE {
    ?topic topico:isAbout dbr:Concert .
    ?topic topico:hasLocation ?location .
     {?topic topico:isAbout ?artist .}
     UNION {?topic topico:hasPerson ?artist .}
    }
  }
 SERVICE <http://dbpedia.org/sparql>
  {SELECT ?artist2 WHERE {
    ?artist2 rdf:type schema:MusicGroup .
    ?artist2 dbp:genre ?musicGenre .
    ?musicGenre dct:subject dbc:Rock_music_genres
    }
  }
FILTER (?artist = ?artist2)
}
```

Figure 6.15. Query of rock musicians and locations of concerts that people have been talking about recently.

it returns one result which is “dbr:Luke_Bryan”, “dbr:Nashville,_Tennessee”.

It is observed that S-Boun-TI makes it possible to process information in Twitter in conjunction with the knowledge existing in LOD, and query that information. Examples have been given of these queries. The queries cannot be issued directly to Twitter or on the raw data that Twitter API returns.

6.4. Observations

This section provides observations about the approach, the results, and problems encountered.

Applying the approach on the public stream (set 11) did not result in any topics. This is due to the the nature of a public stream. Since the tweets are not collected with keywords such as “Toni Braxton” and “concert”, the set does not have dominating topics. Several elements are identified in the public stream, and several relationships (edges) are established in the graph among them. However, these are removed in the edge removal phase. Before coming to a conclusion about public streams, the approach is applied on tweet sets gathered from public stream API starting in 2016-12-04T08:39:49Z and 2017-02-20T19:29:55Z of 5,854 and 6,025 tweets respectively to see if similar results are experienced. These sets are not related to the time of the first public set. Similar results were observed. When this approach is applied to public sets, it did not result in any topics. For future work regarding public streams, the effects of lowering thresholds and using larger numbers of posts could be investigated.

One feature of the method presented here is the maximal cliques application. This strict method has been chosen because with previous techniques that were tested during this study it was possible for unrelated elements to appear in a topic. The previous methods put strongly related elements into one topic. In each step, another element was added according to its cumulative relation weight to the elements in already existing topics. If the cumulative relation weight was above a threshold, then the element was added to that topic. In this way, correct topics were obtained, but incorrect topics were also observed that should have been eliminated. For example, the topic {Donald Trump, Hillary Clinton, Now, Question, Answer, Tax}, which is obtained by this method, is not correct. If the tweets that make up this topic are inspected, it is observed that people were talking about Donald Trump ignoring a question. The subject of the question it is claimed that he ignored was not present in the tweet set. Thus, the correct form of this topic would be {Donald Trump, Now, Answer, Question}. However, the high relationship weight among “Tax”, “Donald Trump” and “Hillary Clinton” caused the cumulative weight to become strong and led to adding “Tax” to the topic. Therefore, relationship weights between each element pair in the topic need to be checked to ensure that each element is related to another within a topic. This is achieved through the maximal cliques when only the edges above τ_e are taken into account. This is the reason for choosing maximal cliques to eliminate incorrect topics due to unrelated element intervention.

Another problem that leads to incorrect topics is issues with the Linked Data. For example, DBpedia can provide wrong information. At the time this thesis was written, the entity "dbr:Women's_rights" has "rdf:type" "foaf:Person", which is incorrect. Another issue is the entity linking tool TagMe. In general it performs well, but incorrect links exist. Incorrect links are caused by the limited context of the posts and lack of corresponding meaning for the spot in Wikipedia. Our approach links some of the unlinked spots using link information from other related posts. However, there are still unlinked spots that may need to be linked. Thus, linking unlinked spots needs further investigation. Entities from other knowledge resources such as Yago [115] and Google knowledge graph [116] are candidates for addressing some of the unlinked spots.

Another way to address unlinked spot expression in the Semantic Web would be to create Semantic Web resources for the unlinked spots that are talked about frequently. The entities, temporal expressions, and observation intervals they are related to would be identified. The only thing not known about this resource is its identity in the real world. If such a technique is applied, this kind of entity becomes referable. This is a way of expressing an unlinked spot as "There is a thing with spot "spot" observed at time interval "starttime-endtime" having related resources {"dbr:Resource$_1$","dbr:Resource$_2$",...,"dbr:Resource$_n$" }".

Location and agent identification needs improvement. S-Boun-TI does not handle unlinked locations and persons. For instance, the post text "conference is starting in the computer engineering building in 5 mins. #compconf2016" indicates a location in the context of an event which is not expressed in LOD. Similarly, a person may not have a resource in LOD, but in the context of tweets he/she is referred to. If such locations and persons were identified, an instance could be created for them. The topic that is created would refer to that instance. Since these locations and persons have relations to other entities in other topics, the linking and disambiguation phase of further identification tasks may utilize these relations. For example, spots such as "conference hall" and "concert hall" can be considered as locations, and before creating an instance, already existing instances can be checked according to several properties such as the words they are used with, the relationships they previously had and the times they appeared before. If there is no previously identified location instance linked to the spot, then a new instance could be created and linked. NLP techniques are helpful to identify persons. For example, spots such as "Michael", "Adam" can indicate persons, and

through similar operations a person instance could be created for each of them or an existing, previously created one could be linked. Location and person identification improvement in S-Boun-TI is a plan for the future. Identification of these types of locations and persons is a further research direction.

In addition to person type, which is focused on, there are other sub-classes of "foaf:Agent" such as "foaf:Group" and "foaf:Organization". At first glance, DBpedia did not express an instance of "foaf:Group" type [117] at the time of writing. However, "foaf:Organization" instances exist in DBpedia. Further study is needed to identify this type of entity according to its context, such as determining whether the entity is an organization or a location. A recent study on the type ranking problem [118] provides insight into this issue. A current data source for locations is DBpedia. Other data sources that provide Semantic Web resources can be added. For example, Geonames has a rich location database which is accessible in LOD. S-Boun-TI could benefit from this database for location identification in the future. On the other hand, it is not easy to identify a location from a short text. There are many locations with the same name, which requires disambiguation.

MusicBrainz [119] is another database which provides artists, albums, songs and their relations in the Semantic Web. Entity linking in this domain is not trivial; songs and album names may match any text piece since they are too numerous and too various.

Topics have emerging behavior. Once a topic emerges, it can be extracted using S-Boun-TI and expressed in the Semantic Web. Topics start with a few main elements and as time passes and people talk about new aspects, new elements come forward. Therefore, there are forms of topics which are related, where each has the same main elements, and where other elements change over time. Relationships among these topics can be expressed using "topico:isAbout" property. Additionally, an extension to $Topic_{\mathcal{O}}$ ontology could be planned to represent the temporal relations of topics. BBC Storyline ontology [120] provides insight into expressing structures that are temporally related and that have properties in common. Expressing and identifying emerging behavior of microblog topics should be addressed in future work.

In some topics, elements or relationships are contributed by specific users or retweets of the same posts. Further investigation is needed to identify these cases and their effect on the quality and the quantity of topics. Meta information that indicates the variety of tweet texts and the users that the elements and relationships are extracted from can be added to topics. Topics can be ranked according to frequencies of relationships and entities, diversity of posters, diversity of words, and diversity of the hashtags of tweets that contribute to elements of the topics.

Certain kinds of events show special usage style, such as using "RIP, dies, death" keywords when someone dies. Special handling of certain types of events can be helpful in identifying elements of topics. Once the type of the event is identified, information such as who is dead, at what age, when, where and why/how he or she died can be inspected in tweets accordingly. This is another future direction of this study.

Performance related issues have not been studied in detail. Working with Linked Data in real-time is challenging. One of the challenges is the long request and response times. While implementing this system, URLs have been cached to overcome this issue. If network calls are cached, computing topics, including maximal cliques, takes about four minutes for an average of 5,700 tweets that are obtained for two minutes in a Linux operating system machine running on Intel Centrino hardware with 2GB RAM. Thus, more optimization is needed in the case of heavy Twitter usage, even if access to external data sources such as Wikipedia, Wikidata, and DBpedia are quickly accessible to the computing process. There are several solutions, including adding more RAM and processing power, and parallelizing processes.

Once the system is real-time, several keywords can be tracked using the streaming API, and these Twitter streams can be transformed into topic streams using S-Boun-TI. Then, these topics can be queried using C-SPARQL [121]. Stream reasoning is possible in these situations.

Another challenge with Linked Data is cross-domain queries. In database research, this concept is also referred to as "federated queries". The problem with federated queries is the inability to efficiently query data that is provided by multiple sources. In conventional

database querying, queries are planned in the same process that has access to all the data and metadata (i.e.indexes, hashes, quantities). Federated querying, however, does not have all information about the data that is stored in other domains. Federated query SPARQL client in Fuseki needs better query planning, especially if the resources in large databases are queried, such as DBpedia and Wikidata. For example, the query in Figure 6.14, which is about the politicians of topics, would have been asked using a federated query similar to the query in Figure 6.15. With two equal join operations, the federated query would be answered. The first part of the query asks for the persons in topics, the second part of the query asks for Wikidata identifiers of persons in DBpedia, and the third part of the query asks for politicians in Wikidata among the persons corresponding to Wikidata identifiers. The first part is joined with the second part, and the second part is joined with the third part. This query is prepared and applied. However, because of the large number of persons in DBpedia, the query does not end in a time suitable for observation. After one hour, the query was manually stopped. However, if the SPARQL client knows which data source is more restricting and should be queried first, the query would take less time. The order of queries presented in Figure 6.14 is planned according to the number of resources that the domains would return. The most restricting query is the first query, which returns only the persons in the topics. The Fuseki server quickly returns 170 distinct persons. Then, DBpedia is queried to retrieve only the identifiers of these persons, which is faster then querying and retrieving all DBpedia persons to the client and matching them with the 170 people. The conclusion from these experiences is that federated queries are becoming an important part of Linked Data, especially if the distributed data is to be utilized. Both the RDF data-providing services and the SPARQL clients (such as Virtuoso [122] and Fuseki) require better handling of federated queries.

6.5. Conclusions

In this chapter, the topic identification approach S-Boun-TI was introduced to identify and express topics of microblog post sets using Semantic Web expressions. A prototype of the proposed approach was implemented and the topics, involving more than one million posts, were examined in detail. Determining topics based on subgraphs of entity co-occurrence graphs using the maximal cliques algorithm was examined to determine whether or not this approach might be too constrained. Likewise, the use of entity linking was examined to

determine if satisfactory results can be obtained. In general, the entity linker TagMe worked well for identifying meaningful parts of posts.

The results are quite promising, producing topics that express the related entities. SPARQL queries produced results that were not obvious in the original posts. Especially queries in conjunction with the data in LOD are quite interesting. All of these observations provide motivation for further progress in this direction by examining alternative approaches for improvement of candidate elements and subgraphs representing topics.

7. RELATED WORK AND COMPARISON

This thesis proposes two novel approaches for identifying topics in collections of microblog posts. The first approach, Boun-TI, produces human readable topics by mapping the microblog posts to Wikipedia article topics. The second approach, S-Boun-TI, produces machine interpretable topics by mapping microblog posts to instances of the "topico:Topic" concept, which is also introduced as part of this thesis.

This chapter presents qualitative comparisons with several aspects of existing approaches, such as the input and output type of the approaches, how they consider microblog posts (collective, or single), and how they approach the topic identification problem in microblogs.

Along with a qualitative comparison of approaches, a quantitative comparison of LDA topics is also provided in this chapter. LDA is one of the widely adopted approaches for topic identification. Section 7.3 gives details about the comparison.

Firstly, the related work of Boun-TI and S-Boun-TI are given. Then, comparisons between Boun-TI and LDA topics and S-Boun-TI and LDA topics are provided. A qualitative comparison between Boun-TI and S-Boun-TI approaches is also provided. Lastly, conclusions about the comparisons are presented.

7.1. Related work of Boun-TI

In existing literature, probabilistic topics are often utilized for identifying topics in collective processing methods. Among these approaches, LDA-based approaches have been proposed by several works [30–34]. LDA assigns related words to sets which are considered as topics. The problem is that these sets do not fully explain the topic. The aim of Boun-TI is to produce topics that can be expressed as Wikipedia articles, which are identifiable resources (unlike words). Wikipedia article titles provide human readable topics. A comparison of LDA and Boun-TI topics is given in Section 7.3.1.

Some approaches have utilized the highly temporal nature of microblog posts to identify topics. [16–22] Approaches of this type are based on collectively processing microblog posts considering the timestamps of words and hashtags. Frequent change in words and hashtags indicates the existence of a topic. To generate possible topics, these approaches either measure the frequency of change of words or create a representative set of posts related to these words.

Lansdall-Welfare *et al.* [6] manually defined keywords for four different classes of moods. In the domain of health, Prieto *et al.* [38] manually specified words and regular expressions, where each group of words and regular expressions indicate a sickness. Parker *et al.* [39] was able to automatically obtain indicative words in the domain of health. These collective processing approaches aim to measure the frequencies of manually defined or automatically obtained words and regular expression matches to determine if there is a related topic. These approaches show the similarity of the manually defined or automatically obtained domain specific keywords to the predefined topic classes. BOUN-TI also measures similarity scores of topics by automatically extracting indicative words of topics. However, unlike these approaches, the topics of BOUN-TI span a wide range of domains that are provided by Wikipedia.

Another collective processing approach was proposed by Sharifi *et al.* [1]. The approach builds a summarizing phrase. The phrase grows towards the left and right of the phrase of interest by considering common consecutive words. BOUN-TI also collectively processes post sets and tries to find topics of sets of posts. However, the topics returned by BOUN-TI are human generated Wikipedia article titles. According to Wikipedia article guidelines, an article's content is represented in the title of the article. The problem is to find the most relevant article. The aim of BOUN-TI is to solve this problem.

Lastly, approaches [10–13] have been proposed that try to enhance microblog posts or short texts by identifying the content of the posts to better identify semantics. TagMe [11] considers Wikipedia titles and internal Wikipedia link structure. The approach by Gattani *et al.* [12] considers data sources such as Yahoo! Stocks and MusicBrainz in addition to Wikipedia titles. In addition to Wikipedia titles and bodies, links and anchors are considered for semantically enhancing single microblog posts by the approach of Meij *et al.* [13]. BOUN-TI considers all posts in the post set, but not single posts, which have limited contexts. Considering the content of Wikipedia article bodies and post sets help identify descriptive

topics, as shown in Section 4.5, which compares aggregation of TagMe results and Boun-TI by giving example topics.

7.2. Related work of S-Boun-TI

S-Boun-TI identifies topics of microblog posts and expresses them in the Semantic Web, unlike the approaches that extract semantic information and express it in the Semantic Web [54–58, 60, 61] from conventional documents such as news documents, meeting reports and blogs. Extracting microblog post topics requires handling microblog texts, which have different characteristics than conventional texts. S-Boun-TI handles user mentions, locations, and temporal expressions that are often referred to in posts and uses them to identify topics.

In the existing literature, short text processing approaches have been proposed [11–13, 23–27] that link spots in short texts to external resources such as Wikipedia articles and DBpedia resources. S-Boun-TI processes a post set by using an entity linker [11], which is a single post processing approach. As observations show, entity linking in short texts is not an easy task and further research is needed. One of the problems with entity linking in short texts is the limited context. S-Boun-TI re-links some of the incorrectly linked entities and links unlinked spots by using the link information in other related posts. S-Boun-TI identifies topics related to post sets, unlike the single post processing approaches.

Approaches that work on single posts and aggregate the information in posts to obtain set level information [6, 9, 23, 38, 39] have been proposed. These approaches often extract information from each post using a variety of methods, such as keyword or regular expression identification and classification of posts to a class (i.e.positive or negative mood)using machine learning techniques. S-Boun-TI relates the identified elements in posts and expresses topics with these elements in the Semantic Web.

Approaches that consider temporal properties of terms and hashtags in microblog posts have been proposed [16–22] to extract information from posts. According to these approaches, change in frequency of words, hashtags or matching regular expressions indicates an emerging topic. The topics are represented as a set of posts or a set of words, unlike S-Boun-TI which

represent topics as Semantic Web resources and uses elements extracted from posts to form topics.

LDA-based approaches are probabilistic topic modeling approaches [30–34, 65] that identify microblog topics. These approaches represent topics as a set of keywords, unlike S-Boun-TI, which represents as Semantic Web resources. A comparison of S-Boun-TI and LDA topics are given in Section 7.3.1.

Approaches have been proposed [62, 63] for semantic representation of single posts. These approaches produce an RDF representation of a single microblog post using meta information such as the author, the creation date, and the entities identified within the text. On the other hand, S-Boun-TI outputs RDF of the topics of a collection of posts by considering relations among the entities within the post set.

7.3. Comparison with LDA

The comparison with LDA is not straightforward due to the inconsistency of the type of outputs of these approaches. Boun-TI produces human readable Wikipedia page titles, S-Boun-TI produces machine interpretable resources mostly consisting of DBpedia URIs, and LDA produces words as topics. To compare these topics, user evaluation is needed. If this is to be done, users must annotate topics by exploring the topics and the post sets and choosing the topics that better represent the post sets.

This thesis provides user evaluation in Chapter 4, which presents the evaluation of Boun-TI by human annotators. While these experiments were being conducted, it was obvious that human annotation is an intensive task; the user effort needed for this kind of work is too great. First of all, the number of posts in each post set is about 6k. For example, in the US presidential election debate datasets, there are 180 post sets of about this size. In Boun-TI experiments, only 30 sets were randomly chosen, and 20 of them were shown to two annotators, with 10 being shown to both annotators. The annotators had to read too many microblog posts (about 6k). This situation is similar. The annotator has to read a very large number of microblog posts, words of LDA topics, and phrases of Boun-TI topics as well as a visualization of the S-Boun-TI topics to understand if a particular topic is

related to a post set and if the elements in LDA and S-Boun-TI topics are related to each other, forming a topic. Deciding on the relatedness of each topic to the input post set took about 1 minute in the Boun-TI experiments. If the 10 topmost topics, and topics of three approaches (30 topics) are considered, about 30 minutes is needed for each post set. If the 20 post sets were to be annotated by a human annotator, about 600 minutes is needed - and this is just for one configuration of the LDA. If multiple configurations of LDA are considered, such as the number of topics, then 10 minutes should be added for each post set and for each configuration. This kind of comparison becomes unfeasible.

Comparing the topics of the approaches gives an idea of the similarity of the results. In this section, this comparison is presented. However, for this experiment, there are challenges as well. To compare topics of LDA, Boun-TI and S-Boun-TI, the elements of topics and the ranking of the topics should be compared. The topic elements of Boun-TI and S-Boun-TI are obtained using external resources such as Wikipedia and DBpedia. However, LDA does not extract topic elements using external resources; it extracts topics from given documents. To obtain topics of post sets using LDA, a document set should be input. One way to do this is to use the same set that is used to identify the topics of Boun-TI and S-Boun-TI, but considering only the posts in the input set would not be fair because Boun-TI and S-Boun-TI use external resources in identification and Boun-TI uses posts of a longer public stream to obtain word distributions of the Twitter environment.

Another way is to use concatenation of the input post set, make a document from this concatenation and identify LDA topics of these documents. If this application is run on the 2016 US presidential election debate datasets that are introduced in Section 6.3.1, it is assumed that each two minute interval of the datasets are made of several topics, and each topic is composed of words. The topics of the specific interval are ranked according to the topic distribution for the document of that interval. Applying LDA on all intervals extracts topic models according to all debate datasets, which makes the obtained topics specific to the domain of the 2016 US presidential election debates. This is different from Boun-TI and S-Boun-TI topics because they are related to the domain of Wikipedia, which spans a wide range of articles of human interest. In the concatenation process of document formation, ordering of microblog posts is not important because LDA takes into account the distribution of words, but not the positions.

The TwitterLDA [123] implementation is used to obtain LDA topics. This implementation is especially tailored for handling tweets. In the application of this implementation on the debate datasets, since there is no prior knowledge of how likely the topics are to be similar and how differentiating the words in topics are, the α, and β parameters are set according to the implemented Java package which are $\alpha = 0.5$ and $\beta = 0.01$. Since the number of topics is an important parameter for LDA in identifying topics, in this experiment, several numbers of topic parameters are set such as 10, 20, ..., 100, and LDA is run with these parameter values. The comparisons for each parameter value are provided in this section.

7.3.1. Comparing Boun-TI and LDA topics

The topics of BOUN-TI are human readable phrases. Considering only the elements in a human readable Wikipedia article title would be insufficient for comparison because LDA provides distribution of many more words as topics. To overcome this limitation of BOUN-TI topics and make the topics of the two approaches comparable, Wikipedia articles of BOUN-TI topics are considered. For comparison, words of BOUN-TI topics are obtained by considering the words of the Wikipedia article bodies. However, not all words in a Wikipedia article should be considered as words of topics. Actually, the words that contribute to the rank of Wikipedia articles should be considered as the words of topics. These words are the common words that both exist in the post set and the Wikipedia article body. Therefore, if the words of BOUN-TI topics are to be extracted, they can be assumed to be words both in the microblog post set and the Wikipedia articles.

The topmost topics and their topmost words are selected, and the similarity between each of them is compared using Jaccard similarity, which is the proportion of the same elements in both topics among the union of all elements (See Equation 7.1).

$$similarity(T_{\text{BOUN-TI}}, T_{LDA}) = \frac{T_{\text{BOUN-TI}} \cap T_{LDA}}{T_{\text{BOUN-TI}} \cup T_{LDA}} \tag{7.1}$$

This algorithm in Figure 7.1 is run and the results are gathered. Table 7.1 gives comparison scores using the Jaccard similarity scores of BOUN-TI topics and LDA topics. The number of topics n are iterated as 10, 20, 30, ..., 100. The average (Avg), standard deviation (σ), and maximum (Max) of values are shown. These are values of all topic comparisons that are obtained for all input post sets.

Input: M ▷ Set of sets of tweets
Input: n ▷ The number of LDA topics
Input: α, β ▷ parameters of LDA
Input: t ▷ paramter for the topmost t topics
Input: w ▷ paramter for the topmost w words of topics
Output: R ▷ Matrix that holds the comparison result values

▷ Get LDA topic model of all post sets (M) for configuration of n topics, α and β
$TopicModel$=ComputeLDA(M,n,α,β)
$i = 0$
while $i < |M|$ **do** ▷ for all tweet sets
    $LDATopics_i$=Rank($TopicModel$,M_i) ▷ Get most relevant LDA topics for M_i
    $BountiTopics_i$=ComputeBounTI(M_i) ▷ Get BOUN-TI topics
    L=TopTopics($LDATopics_i$,t) ▷ Ranked list of top t LDA topics
    B=TopTopics($BountiTopics_i$,t) ▷ Ranked list of top t BOUN-TI topics
    $j = k = 0$
    while $j < |L|$ **do**
        W_L=getLDATopicWords(L_j,w) ▷ Get top w words of LDA topic L_j
        while $k < |B|$ **do**
            W_B=getBountiTopicWords(B_k,w) ▷ Get top w words of BOUN-TI topic B_k
            $R_{i,j,k}$=JaccardSimilarity(W_L,W_B) ▷ Save Jaccard Similarity result
            $k = k + 1$
        end while
        $j = j + 1$
    end while
    $i = i + 1$
end while
return R

Figure 7.1. Comparison algorithm of BOUN-TI and LDA topics

Table 7.1. Comparison of BOUN-TI topics with LDA topics. The topics are compared with Jaccard similarity. The top 5, 10, and 30 words are considered for similarity checking. ($w = 5, 10, 30$) Topmost 10 ranked topics are compared ($t = 10$). The results are given according to the number of topics of LDA. ($n = 10, 20, ..., 100$). To obtain LDA topics, $\alpha = 0.5$ and $\beta = 0.01$. The minimum values are all zeros for all of the rows.

	$w = 5$			$w = 10$			$w = 30$		
n	Avg	σ	Max	Avg	σ	Max	Avg	σ	Max
10	0.14	0.12	0.66	0.15	0.09	0.66	0.11	0.05	0.50
20	0.19	0.14	1.0	0.16	0.09	0.66	0.12	0.04	0.39
30	0.19	0.14	1.0	0.15	0.09	0.53	0.11	0.05	0.39
40	0.18	0.14	1.0	0.16	0.09	0.53	0.11	0.05	0.42
50	0.17	0.13	1.0	0.15	0.09	0.66	0.11	0.05	0.36
60	0.16	0.14	1.0	0.14	0.09	0.53	0.10	0.05	0.39
70	0.15	0.14	1.0	0.14	0.09	0.66	0.11	0.05	0.36
80	0.14	0.13	1.0	0.14	0.09	0.53	0.10	0.05	0.42
90	0.15	0.13	1.0	0.13	0.08	0.53	0.11	0.05	0.42
100	0.14	0.13	1.0	0.13	0.09	0.53	0.10	0.05	0.39

One observation is that the number of topics of LDA does not effect the results. This is due to the number of words that are common in every topic of LDA topics, such as “hillary”, “debate” and “donald”. One of the reasons for the lower similarity scores is the source of the topic terms. BOUN-TI topics do not have some Twitter-specific terms such as “vpdebate”, while LDA topics have, which results in lower scores.

Another observation is that the number of words of topics (w) that are taken into account in the similarity comparison of BOUN-TI and LDA topics effects the results. The higher similarity scores are obtained with a smaller number of words of topics. While w increases, σ decreases, which suggests that the smaller number of words provides diverse comparison scores, while a higher number of words provides less diverse scores. For $w = 5$, the same topics are observed (similarity score of 1). While w increases, the maximum similarity score (Max) decreases, which suggests that the more words that are considered as topics, the

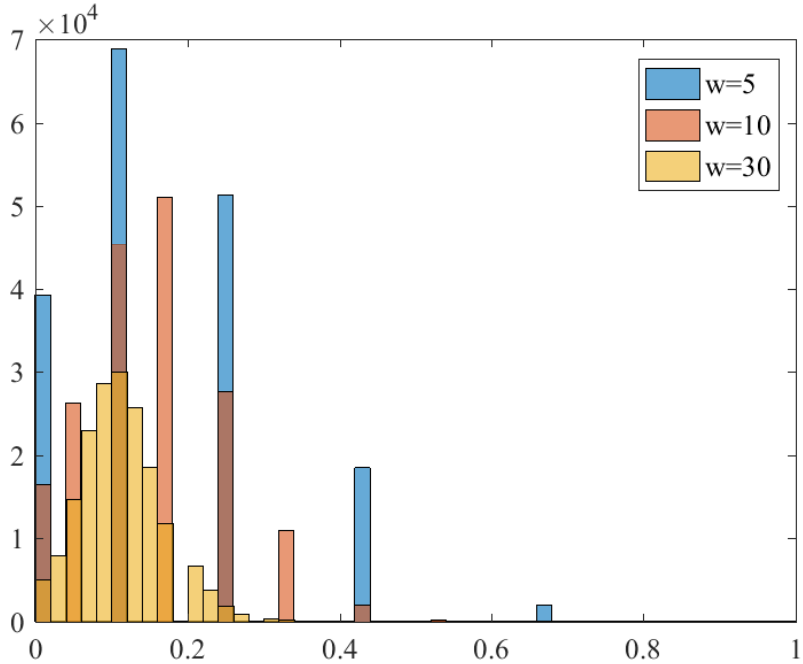


Figure 7.2. The histogram of all comparison scores of BOUN-TI and LDA topics with respect to w. The x axis indicates the scores, and the y axis indicates the number of times a score is observed.

less similar they are.

Figure 7.2 gives the histogram of all scores obtained in the comparison of BOUN-TI and LDA topics. It is also observed from the histogram that lower w scores result in more similar scores.

Figure 7.3 shows a heatmap of average comparison scores of BOUN-TI and LDA topics over the 2-minute intervals of the four debates. It is observed that the second presidential debate provides the most similar topics. The vice presidential debate provides the least similar topics. This is due to the query terms related to the vice presidential debates, such as "vpdebate" existing in the LDA topics but not existing in the BOUN-TI topics. The values of n that give higher scores are $n = 20, 30, 50, 60$.

So far, the quantitative properties of the comparison results have been presented. In the following section, some of the topics that are compared are presented, and a qualitative

analysis is provided. Since it is not a trivial task to understand each topic of LDA, a portion of the dataset is manually inspected, and only some sample topics are provided. One of the sample topics, obtained in the second presidential debate for $w = 5$ is the list of words [clinton, hillary, bill, debate, women] ranked according to the words' scores. For these topics, while the words "clinton, hillary, and debate" are common words, "bill" and "women" are the distinctive words. Both the LDA topic and the BOUN-TI topic provide these words. If other topics of $w = 5$ configuration are inspected, it is observed that high similarity scores are obtained because of the large portion of words that are common, such as "hillary", and "donald".

If $w = 10$ configuration is inspected, it is observed that a comparison score of 1 is not obtained. One of the topmost comparison scores is obtained in the 38th interval (76th minute of the first presidential debate). The fifth ranked BOUN-TI topic and 2nd ranked LDA topic has the highest comparison score, which is 0.66. The topmost ten words of the BOUN-TI topic are [debate, hillary, clinton, debates, obama, iraq, candidates, presidential, president, war] while the topmost ten words of the LDA topic are [hillary, debate, debatenight, clinton, trump, debates, president, iraq, war, obama]. Other topics of $w = 10$ are also manually inspected. Upon observation, it can be concluded that the BOUN-TI and LDA topics of $w = 10$ provide a higher number of matching words, but lower comparison scores. Upon manual inspection, it is also observed that $w = 30$ provides an even higher number of matching words, but the lowest comparison scores that can be seen overall in Figure 7.2.

7.3.2. Comparing S-Boun-TI and LDA topics

The topic elements of S-BOUN-TI are linked entities (URIs) that are defined in Linked Data. Equality comparison of URIs and the words in topics will result in low similarity. To address this problem, the words in entity names are considered for comparison. Therefore, the topics are converted to word sets to compare with sets of LDA topics. In this study, only string and substring matches are used, which gives an idea of the similarities of the topics.

The LDA topics are ranked according to the topic distribution for the given post set. However, S-BOUN-TI topics are not ranked. Since the S-BOUN-TI topic lengths may vary (for example, there are topics with three elements and topics with seven elements), the topics

with the most words are used for comparison. Therefore, the S-Boun-TI topics are ranked according to the number of words they provide.

For the comparison, Jaccard similarity is used, which was shown in Equation 7.1. Two different versions are applied in Jaccard similarity comparison. The first version is the original Jaccard similarity which is named "exact match" Jaccard similarity. The second version considers a substring match between two words of both sides as a case of equality for intersection and union computations. This version is named "substring match" Jaccard similarity. In all Jaccard similarity computations, the same number of words are considered for both sides of the comparison. Therefore, if the number of words (v) that an S-Boun-TI topic provides is less than the number of words that an LDA topic provides, only the top v LDA topic words are considered.

The algorithm in Figure 7.4 is run and the results are obtained. Table 7.2 gives a comparison using Jaccard similarity scores of the S-Boun-TI topics with LDA topics using Jaccard Similarity. For the comparison results in the table, the number of words considered for similarity is equal to the number of words the S-Boun-TI topic provides. The topmost 10 ranked topics are compared. ($t = 10$). The results are given according to the number of topics of LDA. ($n = 10, 20, ..., 100$). To obtain LDA topics, $\alpha = 0.5$ and $\beta = 0.01$. The S-Boun-TI topics are ranked in descending order according to the number of words they provide.

Similarity scores of S-Boun-TI topics and LDA topics tend to be higher than the similarity scores of Boun-TI and LDA topics. One reason for this is that the number of words of the comparisons are determined by the number of words in the S-Boun-TI topic. Typically, the number of words of an S-Boun-TI topic is five to ten. In other words, the lengths of the topics in these comparisons are often lower than the Boun-TI comparisons which are fixed as five, ten and thirty. This results in higher comparison scores, because manual observations on the topics and scores reveals that the topmost words tend to be more similar.

The "Exact match" column shown in Table 7.2 indicates the original Jaccard similarity. The words are strings, and the strings are checked for equality. If they are equal, then the two

Table 7.2. Comparison of S-Boun-TI topics with LDA topics. The topics are compared using Jaccard similarity.

	Exact match			Substring match		
n	Avg	σ	Max	Avg	σ	Max
10	0.16	0.11	0.63	0.33	0.22	1.0
20	0.19	0.11	0.71	0.36	0.22	1.0
30	0.18	0.11	0.77	0.35	0.20	1.0
40	0.20	0.11	0.75	0.37	0.21	1.0
50	0.20	0.11	0.77	0.38	0.22	1.0
60	0.19	0.11	0.80	0.36	0.21	1.0
70	0.17	0.11	0.71	0.33	0.21	1.0
80	0.18	0.11	0.77	0.36	0.22	1.0
90	0.18	0.10	0.77	0.36	0.21	1.0
100	0.17	0.11	0.77	0.34	0.21	1.0

elements are considered to be the same. The "Substring match" column shown in Table 7.2 indicates that the comparison is based on the rule that two elements are considered to be the same if one of them is a substring of the other.

The "Substring match" scores are higher than the "exact match" scores. This is an expected result. "Substring match" mostly considers the plural and singular forms of words such as "election, debate" and "elections, debate", which are actually the same. "Substring match" resulted in similarity scores of 1. This indicates that there are topics of S-Boun-TI which are the same as a considered topmost word of an LDA topic.

Figure 7.5 shows the histogram of all comparison scores. It is observed that substring match-based Jaccard similarity results in higher comparison scores. Full match of topics (score of 1) are observed in this case.

Figure 7.6 shows a heatmap of average comparison scores of S-Boun-TI and LDA topics over the 2-minute intervals of the four debates. If the top scoring topics are investigated, it is observed that the maximum score returned by exact matching is 0.8. In this example,

the words of the S-Boun-TI topics are [donald, trump, tim, kaine, mike, pence, vladimir, putin, russia] and the words of the LDA topics are [vpdebate, pence, trump, kaine, putin donald, mike, tim, russia]. The S-Boun-TI topic which gives [hillary, clinton, donald, trump, debate] and the LDA topic [hillary, trump, debatenight, debates, donald, clinton] score is one of the highest among the comparisons. These comparison scores are quite satisfying. Manual inspection of the topics and the scores also confirmed that the topics that receive high comparison scores are related. However, there are topics that are related but could not be identified by the comparison method. For instance, the term "obamacare" is often referred to in tweets. However, S-Boun-TI topics refer to the resource of the term, which is "dbr:Patient_Protection_and_Affordable_Care_Act". Since the algorithm extracts terms from the name of the resource, "obamacare" does not match, resulting in a lower comparison score.

7.4. Comparing Boun-TI and S-Boun-TI topics

While working on the two approaches, many topics are manually observed. In this section, a qualitative analysis of these observations, as well as the strength and required time for completion of both approaches are provided.

When people talk about events such as protests, conferences, concerts etc, they use similar terms. For example, political campaigns often refer to similar issues. For these reasons, for post sets that are location- or event-centric, Boun-TI experienced challenges. Furthermore, many topics discussed in politics come up again and again over time, so the contents of microblog posts will have similarities from one campaign to another. Protests also have some similar traits. Thus, when the conversation on Twitter is compared with Wikipedia content, it is easy to confuse the campaign of Hillary Clinton in 2008 and 2016 or protests of the North Dakota pipeline with Arab Spring, Gezi Park, or anti-Iraq War. When compared with Boun-TI, S-Boun-TI performed much better in this context, since S-Boun-TI focuses on persons, temporal aspects and locations. Further6more, S-Boun-TI expects relevant entities to co-occur across contributions, which provides the appropriate context.

The year problem, which is mentioned in Section 4.6, was also encountered while working on the development of S-Boun-TI. Entity linking approaches based on Wikipedia and short messages must check if the message specifically refers to a year. For example, the en-

tity linker that is used by S-BOUN-TI links the spot "presidential debate" to "United States presidential debates, 2012" in some of the tweets, while the actual context is 2016. This issue is resolved as explained in Section 6.2.1.

BOUN-TI topics often provide several term and concepts. However, if the microblog post set does not have a dominant topic that can be represented by a Wikipedia page title, then identification of less detailed topics like single concepts and proper nouns such as "Catholic Church" and "Iran" is desired, which is currently not part of its behavior. S-BOUN-TI, on the other hand, does not have this problem because the elements of topics are linked entities. On the other hand, S-BOUN-TI topics provide structures that present related elements, but not human readable phrases like Wikipedia titles provide.

While BOUN-TI topics do not emphasize agents, locations, or temporal expressions, S-BOUN-TI topics focus specifically on these types. These elements can be observed from the BOUN-TI topics, which are Wikipedia titles, but the information that can be gathered from BOUN-TI topics is limited. For instance, Wikipedia articles mostly have year referrals. On the other hand, S-BOUN-TI topics provide the observation interval of the topic, the year, month, or season, and relative expressions such as today, tomorrow. The relative expressions can be processed further to yield more information about the temporal aspects. For instance, a topic which has an observation interval of 29.01.2017 and the temporal expression "topico:Tomorrow" is actually about 30.01.2017. If temporal reasoning is added, a query about 30.01.2017 would consider this topic.

Another difference of BOUN-TI and S-BOUN-TI is about the utilization of topics. While people can read and interpret the BOUN-TI topics, in addition to interpreting S-BOUN-TI topics people can query the topics semantically, since the topics and their elements are provided in a semantically represented structure. Examples of some of these queries have been provided in Section 6.3.5.

7.5. Conclusions

The outputs and the approaches make comparisons between the BOUN-TI topics and LDA topics, as well as the S-BOUN-TI topics and the LDA topics challenging tasks. While

BOUN-TI provides URLs of Wikipedia articles represented by human readable phrases as topics, S-BOUN-TI provides topics as machine interpretable resources that include references to other machine interpretable resources mostly in Linked Data. It is also non-trivial to decide on the parameters of LDA which are the α, β and the number of topics (n).

Another option using existing gold standard datasets for comparison is also investigated. The study by Aiello *et al.* [124] aims to provide gold standard datasets about topics of tweets. They provide election related datasets from 2012 along with datasets from the Football Association Challenge Final Cup. The datasets are manually annotated by users. The users reported the topics of the datasets. To gain insight into the usefulness of these datasets in computing the success of the approaches, they are manually investigated. The observations revealed that, for over 30M tweets, a few topics (about 2, 3, or 4) are provided by the users. Regardless of the correctness of these topics, they are not suitable for comparison with the topics of BOUN-TI and S-BOUN-TI because of the cardinalities. It is manually observed that the topics obtained by S-BOUN-TI and BOUN-TI often provide few words of the topics in datasets, which will result in high scores. Another reason for not applying the approaches on these datasets is the rate limit of Twitter in retrieving tweets, and Twitter policies about distributing tweets. It is challenging to retrieve all the tweets in the dataset to apply on BOUN-TI and S-BOUN-TI, which takes days.

Another option is to use ranking comparison methods [125, 126]. Ranking comparison is often applied to rankings of the same type of items. To apply these operations, the ranked items, which are topics, should be the same but in a different order. However, the topics are not guaranteed to be the same in the context of this comparison.

In these experiments, topmost ranked t topics are compared with each other. To compare BOUN-TI and S-BOUN-TI topics with LDA, they are converted to word sets, which are suitable for set-similarity comparison. Since the topics are actually identifiers, this operation is challenging. After comparison, common words such as "trump, hillary etc" and a few words related to topics like "healthcare, and tax" are observed. For some specific cases, such as considering the topics as five words, or considering the LDA topics to be the same length as the S-BOUN-TI topic, the comparisons resulted in the maximum score of 1. For other cases, on average, the comparisons did not result in high scores. One reason is that all of the

topmost ten topics are compared with all of the topmost ten topics, which makes a hundred comparison results. It would be unrealistic to expect all topics to be similar. The problem in this task is the lack of knowledge about the topic pairs to compare. Therefore, all of them are compared. Another problem is the transformation of Boun-TI and S-Boun-TI topics to word sets. Boun-TI topics do not include Twitter-specific words such as "debatenight" and event specific terms such as "vpdebate". This situation lowers the similarity scores. One more problem is the unsuitability of S-Boun-TI topics for representing words, such as in the case of "Patient protection and affordable care act" which are referred to in various ways on Twitter such as "obamacare" and "aca" (affordable care act). The regularity of naming of the S-Boun-TI topic elements lowers the matching probability of the words, which results in lower similarity scores. The last problem is the lack of knowledge about the number of topics and other LDA parameters. While in this study, the effect of the number of topics has been investigated, further investigation is needed.

All in all, it is observed that, Boun-TI and S-Boun-TI provide fewer comparison scores that can be interpreted as similar and a high number of comparison results that can be interpreted as partially similar. In string comparisons, additionally, NLP techniques such as stemming and Levenshtein distance computation could be applied to words in topics as part of future work. This would improve matching similar elements which makes the comparison more fine-grained. In these comparisons, however, this has not been applied.

In the comparison of Boun-TI and S-Boun-TI with each other, it is observed that they each have both pros and cons. For machine interpretability, S-Boun-TI is suggested. For understanding of the topics of a post set as human, visualization of S-Boun-TI topics and Boun-TI are suggested. Further processing of S-Boun-TI topics is easier than the Boun-TI topics since S-Boun-TI topics are structured and the elements of the topics are URIs in Linked Data.

When the existing work in this field and the comparison results are considered, it can be concluded that Boun-TI and S-Boun-TI provide topics of microblog post sets and express them differently than in previous approaches. It is a challenging task to compare the results of these approaches. In this section, comparison with one of the widely utilized topic identification methods, LDA, is presented.

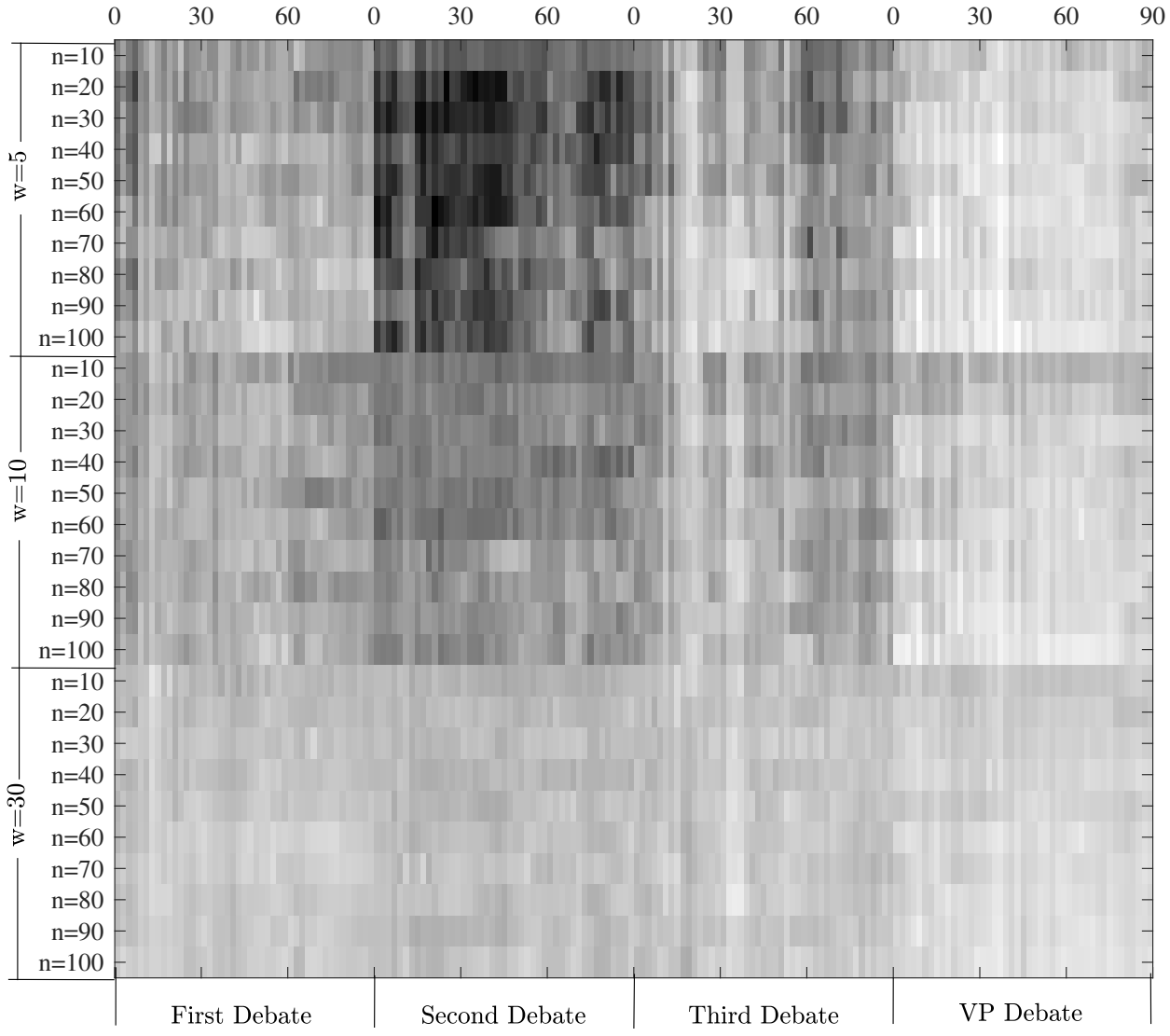


Figure 7.3. Heatmap of comparison scores of BOUN-TI and LDA topics over the four debate intervals (2-minute intervals of each 90 minute debate). The scores are the average of the topmost five BOUN-TI and LDA topics. Darkness indicates high similarity score. The darkest areas indicate a score of 0.5.

Input: M ▷ Set of sets of tweets
Input: n ▷ The number of LDA topics
Input: α, β ▷ parameters of LDA
Input: t ▷ paramter for the topmost t LDA topics
Output: R ▷ Matrix that holds the comparison result values

▷ Get LDA topic model of all post sets (M) for configuration of n topics, α and β
$TopicModel$=ComputeLDA(M,n,α,β)
$i = 0$
while $i < |M|$ **do** ▷ for all tweet sets
    $LDATopics_i$=Rank($TopicModel$,M_i) ▷ Get most relevant LDA topics for M_i
    $SBounTITopics$=ComputeSBounTI(M_i) ▷ Get S-Boun-TI topics
    L=TopTopics($LDATopics_i$,t) ▷ Ranked list of top t LDA topics
    S=TopTopics($SBounTITopics$,t) ▷ Ranked list of top t S-Boun-TItopics
    $j = k = 0$
    while $j < |L|$ **do**
        while $k < |S|$ **do**
            W_S=getSBountiTopicWords(S_k) ▷ Get words of S-Boun-TI topic S_k
            W_L=getLDATopicWords(L_j,W_S) ▷ Get top $|W_S|$ words of LDA topic L_j
            $R_{i,j,k}$=JaccardSimilarity(W_L,W_S) ▷ Save Jaccard Similarity result
            $k = k + 1$
        end while
        $j = j + 1$
    end while
    $i = i + 1$
end while
return R

Figure 7.4. Comparison algorithm of S-Boun-TI and LDA topics

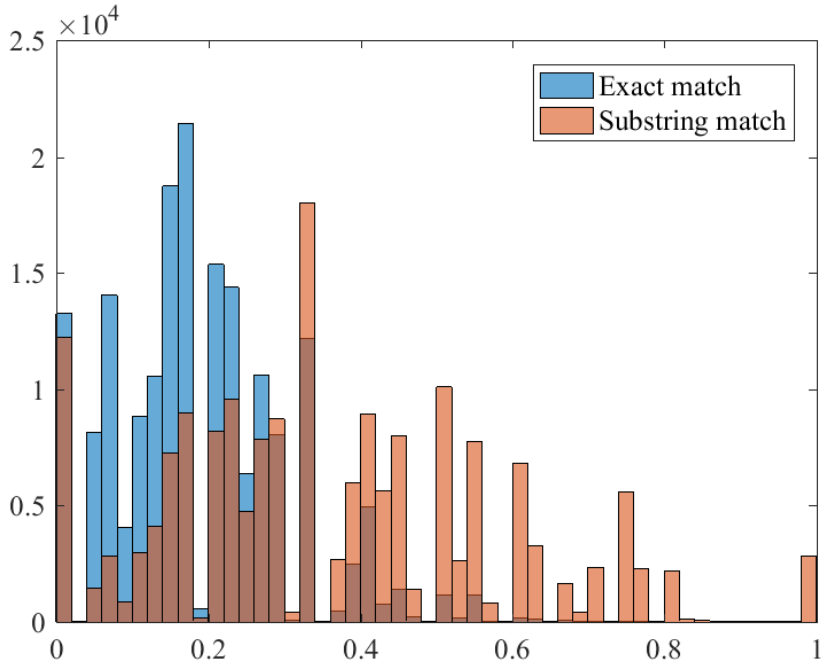


Figure 7.5. The histogram of all comparison scores of S-Boun-TI. The exact match and substring match are given. The x axis indicates the scores, and the y axis indicates the number of times a score is observed.

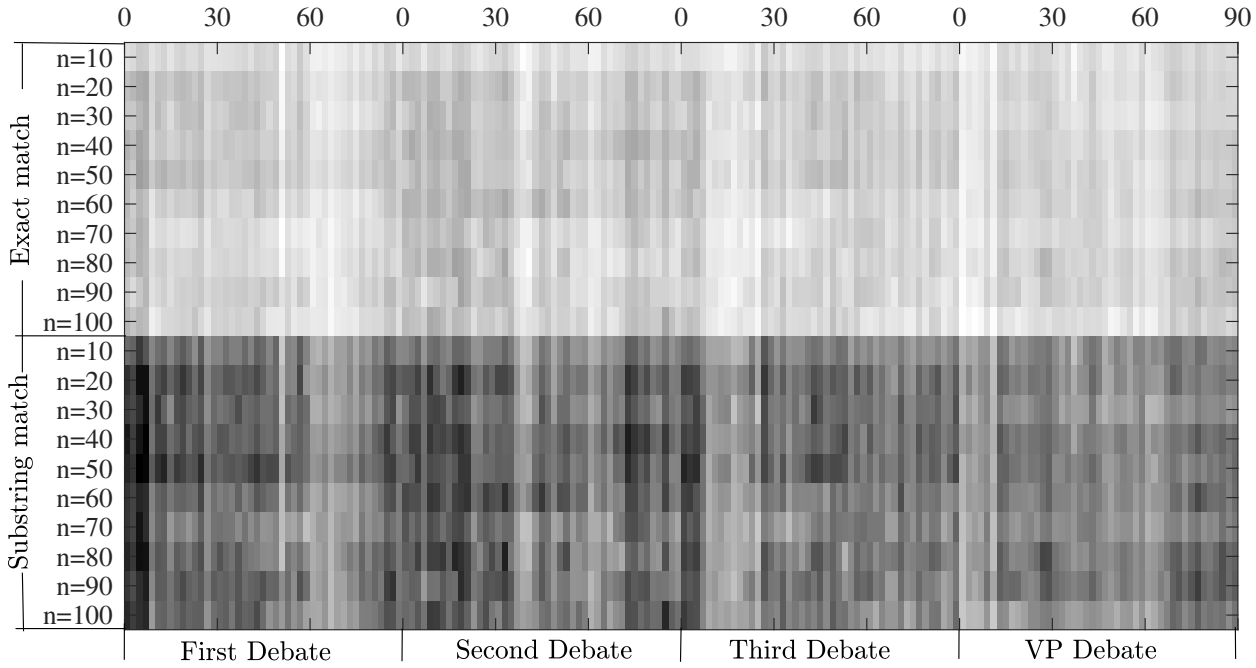


Figure 7.6. Heatmap of comparison scores of S-Boun-TI and LDA topics over the four debate intervals (2-minute intervals of each 90 minute debate). The scores are the average of the comparison scores of S-Boun-TI and LDA topics in the interval. Darkness indicates high similarity score. The darkest areas indicate a score of 0.5.

8. DISCUSSION AND FUTURE WORK

In sections 4.4, 6.4 and in Chapter 7, several observations about the topics of both approaches have been presented along with discussion and future directions for improving each of them. In this chapter, a discussion and possible future directions of BOUN-TI and S-BOUN-TI in conjunction with one another are provided. The future improvements introduced in this chapter are based on the idea of using one approach to improve results of the other.

One of the problems in the application of BOUN-TI is the case of too many weak topics. Microblogs may contain many subjects that are talked about by few people. Sometimes, this results in post sets without strong topics. When this happens, in some cases, BOUN-TI may encounter problems. Addressing this problem is one goal for future work on BOUN-TI. This can be achieved by raising the importance of Wikipedia titles in computations. Because Wikipedia often has an article for single concepts with shorter names such as "Catholic Church", another solution would be using entity linkers. Post by post, an entity linker is run and the titles of linked Wikipedia articles are given as output if scores of BOUN-TI topics are not above a specific threshold. These operations suggest the idea of a topic identification method that firstly tries to give BOUN-TI topics over a confidence score as output. If the topics scores are not sufficiently confident, then entities that are linked are output as topics.

Another research direction that might solve the problem of many weak topics is gathering Wikipedia titles as topics from S-BOUN-TI topics. This is possible by processing the topics of S-BOUN-TI. One way is to compare S-BOUN-TI topics with Wikipedia page titles. A title that refers to most of the elements in an S-BOUN-TI topic can be ranked high. In addition to page titles, page bodies can be compared with S-BOUN-TI topics. To improve the comparison process, entity linking can be applied to Wikipedia page titles and bodies, and in addition to texts, the entities can also be compared. If this is applied, the incorrect topic "Economy of Canada" (which is given as an example in the BOUN-TI conclusions presented in Section4.6) would be eliminated. In this context, applying an LDA method that identifies related word sets similar to S-BOUN-TI, which provides related element sets, can be investigated. The comparisons in Section 7.3 showed that applying LDA is not trivial. Even so, if the LDA word sets are obtained, they can be used to identify Wikipedia article

titles by comparing the article contents and the titles.

Topic$_{\mathcal{O}}$ represents temporal aspects using two different structures. One of them is the observation interval of the topics. The other is relative expressions such as “Now”, “Today”, “Tomorrow”, and other expressions such as year, month and season names. One use case is to directly utilize these expressions in the processing of topics. Another use case is to employ temporal reasoning of relative expressions by adding rules to the knowledge base.

In this study, temporal expressions are properties of topics. However, the usability of these expressions has not been covered. Temporal reasoning in topics requires further investigation. The temporal reasoner that works on S-Boun-TI topics needs to consider the time interval of topics. Daylight saving times and various local times of the users who contribute to the elements should be considered in temporal reasoning. Temporal processing and reasoning tasks should be addressed in future work.

9. CONCLUSION

In this thesis, two topic identification methods have been introduced. The methods extract topics from sets that have many posts. The first approach, BOUN-TI, which identifies topics as Wikipedia articles, gives encouraging results and is effective in identifying topics. The second approach, S-BOUN-TI, identifies semantically structured topics through the use of the $Topic_{\mathcal{O}}$ ontology (also introduced as part of this work).

BOUN-TI identifies elements of topics as distinguishing words of the post set, and using these words, it identifies related Wikipedia articles. Wikipedia articles are often effective since they are also kept up-to-date by user generated contributions with material relevant to topics of interest. It also has the benefit of being human readable, which is an issue often left as an extra step in approaches that yield keyword sets or post sets as topics. The titles of Wikipedia pages efficiently represent the topics. The challenges of this approach stem from deriving *tf-idf* scores based on a bag of words approach, which does not retain the relationships between the words. This is particularly problematic when handling microblog post sets that have a strong presence of posts about multiple topics.

In the approach, S-BOUN-TI, the topic elements correspond to linked entities. The relationships among entities are based on co-occurrence in posts. This approach yields higher quality topic elements as well as retaining the relationships among the elements. Not only are the elements more informative, but since they are linked to semantic entities in DBpedia and queryable with SPARQL, queries reveal information beyond the terms explicitly used in the posts.

In conclusion, both approaches provide compelling results. The promise of these results should motivate the continuation of both approaches as well as their combination, especially to yield human readable topics in Wikipedia.

Finally, in order to aid further work related to this thesis, as well as potential future research, several resources are provided. These resources include implementations for retrieving posts from Twitter, BOUN-TI, and S-BOUN-TI; other coding for utilization of several

systems such as DBpedia, Wikidata, TagMe and other tools like iGraph library for R. Post sets (post identifiers) that are used in the evaluations of Boun-TI and S-Boun-TI; over 9M topics that are identified using S-Boun-TI; topics of Boun-TI that are extracted from 180 sets; as well as annotator choices and other related resources that may be needed to reproduce our tests are provided.

REFERENCES

1. Sharifi, B., M.-A. Hutton and J. K. Kalita, "Experiments in Microblog Summarization", *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pp. 49–56, IEEE Computer Society, Washington, DC, USA, 2010.

2. Twitter, *Twitter*, 2017, https://twitter.com/, accessed at April 2017.

3. Brain, S., *Twitter Statistics*, 2016, http://www.statisticbrain.com/twitter-statistics/, accessed at April 2017.

4. Eisenstein, J., "What to do about bad language on the internet", *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 359–369, Association for Computational Linguistics, Atlanta, Georgia, June 2013.

5. Hu, M., S. Liu, F. Wei, Y. Wu, J. Stasko and K.-L. Ma, "Breaking News on Twitter", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2751–2754, ACM, New York, NY, USA, 2012.

6. Lansdall-Welfare, T., V. Lampos and N. Cristianini, "Effects of the Recession on Public Mood in the UK", *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, pp. 1221–1226, ACM, New York, NY, USA, 2012.

7. Boukes, M. and D. Trilling, "Political relevance in the eye of the beholder: Determining the substantiveness of TV shows and political debates with Twitter data", *First Monday*, Vol. 22, No. 4, 2017.

8. Bennett, T. M., *Marketing Strategies: How Small Restaurant Businesses use Social Media*, Ph.D. Thesis, Walden University, 2017.

9. Sakaki, T., M. Okazaki and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-time

Event Detection by Social Sensors", *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pp. 851–860, ACM, New York, NY, USA, 2010.

10. Ferragina, P. and U. Scaiella, "TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities)", *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 1625–1628, ACM, New York, NY, USA, 2010.

11. Ferragina, P. and U. Scaiella, "Fast and Accurate Annotation of Short Texts with Wikipedia Pages", *IEEE Software*, Vol. 29, No. 1, pp. 70–75, 2012.

12. Gattani, A., D. S. Lamba, N. Garera, M. Tiwari, X. Chai, S. Das, S. Subramaniam, A. Rajaraman, V. Harinarayan and A. Doan, "Entity Extraction, Linking, Classification, and Tagging for Social Media: A Wikipedia-based Approach", *Proc. VLDB Endow.*, Vol. 6, No. 11, pp. 1126–1137, Aug. 2013.

13. Meij, E., W. Weerkamp and M. de Rijke, "Adding Semantics to Microblog Posts", *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pp. 563–572, ACM, New York, NY, USA, 2012.

14. Tran, T., N. K. Tran, T. H. Asmelash and R. Jäschke, "Semantic Annotation for Microblog Topics Using Wikipedia Temporal Information", *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 97–106, 2015.

15. Go, A., R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision", *CS224N Project Report, Stanford*, Vol. 1, No. 12, 2009.

16. Alvanaki, F., S. Michel, K. Ramamritham and G. Weikum, "See What's enBlogue: Real-time Emergent Topic Identification in Social Media", *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, pp. 336–347, ACM, New York, NY, USA, 2012.

17. Cataldi, M., L. Di Caro and C. Schifanella, "Emerging Topic Detection on Twitter Based

on Temporal and Social Terms Evaluation", *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, MDMKDD '10, pp. 4:1–4:10, ACM, New York, NY, USA, 2010.

18. Chen, Y., H. Amiri, Z. Li and T.-S. Chua, "Emerging Topic Detection for Organizations from Microblogs", *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pp. 43–52, ACM, New York, NY, USA, 2013.

19. Kasiviswanathan, S. P., P. Melville, A. Banerjee and V. Sindhwani, "Emerging Topic Detection Using Dictionary Learning", *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 745–754, ACM, New York, NY, USA, 2011.

20. Marcus, A., M. S. Bernstein, O. Badar, D. R. Karger, S. Madden and R. C. Miller, "Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 227–236, ACM, New York, NY, USA, 2011.

21. Mathioudakis, M. and N. Koudas, "TwitterMonitor: Trend Detection over the Twitter Stream", *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pp. 1155–1158, ACM, New York, NY, USA, 2010.

22. Trilling, D., "Two different debates? Investigating the relationship between a political debate on TV and simultaneous comments on Twitter", *Social Science Computer Review*, Vol. 33, No. 3, pp. 259–276, 2015.

23. Schulz, A., C. Guckelsberger and F. Janssen, "Semantic Abstraction for generalization of tweet classification: An evaluation on incident-relatedtweets", *Semantic Web*, Vol. 8, No. 3, pp. 353–372, 2016.

24. Waitelonis, J. and H. Sack, "Named Entity Linking in #Tweets with KEA", *Proceedings of 6th workshop on 'Making Sense of Microposts', Named Entity Recognition and Linking*

(NEEL) Challenge in conjunction with 25th International World Wide Web Conference (WWW), pp. 61–63, April 2016.

25. Brian, P. T.-T. H. H. and W. B. R. H. C. Hayes, "Kanopy4Tweets: Entity Extraction and Linking for Twitter", *Proceedings of 6th workshop on 'Making Sense of Microposts', Named Entity Recognition and Linking (NEEL) Challenge in conjunction with 25th International World Wide Web Conference (WWW)*, pp. 64–66, April 2016.

26. Greenfield, K., R. Caceres, M. Coury, K. Geyer, Y. Gwon, J. Matterer, A. Mensch, C. Sahin and O. Simek, "A reverse approach to named entity extraction and linking in microposts", *Proceedings of 6th workshop on 'Making Sense of Microposts', Named Entity Recognition and Linking (NEEL) Challenge in conjunction with 25th International World Wide Web Conference (WWW)*, pp. 67–69, April 2016.

27. Caliano, D., E. Fersini, P. Manchanda, M. Palmonari and E. Messina, "UniMiB: Entity linking in tweets using Jaro-Winkler distance, popularity and coherence", *Proceedings of 6th workshop on 'Making Sense of Microposts', Named Entity Recognition and Linking (NEEL) Challenge in conjunction with 25th International World Wide Web Conference (WWW)*, pp. 70–72, April 2016.

28. Derczynski, L., D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak and K. Bontcheva, "Analysis of named entity recognition and linking for tweets", *Information Processing & Management*, Vol. 51, No. 2, pp. 32 – 49, 2015.

29. Xu, T. and D. W. Oard, "Wikipedia-based topic clustering for microblogs", *Proceedings of the American Society for Information Science and Technology*, Vol. 48, No. 1, pp. 1–10, 2011.

30. Diao, Q., J. Jiang, F. Zhu and E.-P. Lim, "Finding Bursty Topics from Microblogs", *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pp. 536–544, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012.

31. Phan, X.-H., L.-M. Nguyen and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections", *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pp. 91–100, ACM, New York, NY, USA, 2008.

32. Ramage, D., S. Dumais and D. Liebling, "Characterizing microblogs with topic models", *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 130–137, AAAI, 2010.

33. Yan, X., J. Guo, Y. Lan and X. Cheng, "A Biterm Topic Model for Short Texts", *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 1445–1456, ACM, New York, NY, USA, 2013.

34. Zhao, W. X., J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan and X. Li, "Comparing Twitter and Traditional Media Using Topic Models", *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pp. 338–349, Springer-Verlag, Berlin, Heidelberg, 2011.

35. Genc, Y., Y. Sakamoto and J. V. Nickerson, "Discovering context: classifying tweets through a semantic transform based on wikipedia", *Proceedings of the 6th international conference on Foundations of augmented cognition: directing the future of adaptive systems*, FAC'11, pp. 484–492, Springer-Verlag, Berlin, Heidelberg, 2011.

36. Petrović, S., M. Osborne and V. Lavrenko, "Streaming first story detection with application to twitter", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–189, Association for Computational Linguistics, 2010.

37. Vitale, D., P. Ferragina and U. Scaiella, *Advances in Information Retrieval: 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings*, chap. Classification of Short Texts by Deploying Topical Annotations, pp. 376–387, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

38. Prieto, V. M., S. Matos, M. Álvarez, F. Cacheda and J. L. Oliveira, "Twitter: A Good Place to Detect Health Conditions", *Plos One*, Vol. 9, No. 1, pp. 1–11, 01 2014.

39. Parker, J., Y. Wei, A. Yates, O. Frieder and N. Goharian, "A Framework for Detecting Public Health Trends with Twitter", *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pp. 556–563, ACM, New York, NY, USA, 2013, http://doi.acm.org/10.1145/2492517.2492544.

40. Wang, H., D. Can, A. Kazemzadeh, F. Bar and S. Narayanan, "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle", *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pp. 115–120, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012.

41. Kouloumpis, E., T. Wilson and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!", *Proceedings of the Fifth International AAAI Conference on Web and Social Media*, pp. 538–541, 2011.

42. Boğaziçi University, *SBounTi topics*, 2017, http://193.140.196.97:3030/topic, accessed at April 2017.

43. Lin, C.-Y., "Knowledge-based Automatic Topic Identification", *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pp. 308–310, Association for Computational Linguistics, Stroudsburg, PA, USA, 1995.

44. Chen, K.-h., "Topic identification in discourse", *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pp. 267–271, Morgan Kaufmann Publishers Inc., 1995.

45. He, X., C. H. Q. Ding, H. Zha and H. D. Simon, "Automatic topic identification using webpage clustering", *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 195–202, 2001.

46. Joshi, M. and N. Belsare, "BlogHarvest: Blog Mining and Search Framework.", *COMAD*, pp. 226–229, 2006.

47. Java, A., X. Song, T. Finin and B. Tseng, "Why We Twitter: Understanding Microblogging Usage and Communities", *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07, pp. 56–65, ACM, New York, NY, USA, 2007.

48. Beal, V., *Twitter Dictionary: A Guide to Understanding Twitter Lingo*, 2016, http://www.webopedia.com/quick_ref/Twitter_Dictionary_Guide.asp, accessed at April 2017.

49. Berners-Lee, T., J. Hendler, O. Lassila *et al.*, "The semantic web", *Scientific american*, Vol. 284, No. 5, pp. 28–37, 2001.

50. Hendler, J., "Web 3.0 Emerging", *Computer*, Vol. 42, No. 1, pp. 111–113, Jan 2009.

51. Bizer, C., J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, "DBpedia - A crystallization point for the Web of Data", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 7, No. 3, pp. 154 – 165, 2009, the Web of Data.

52. Abele, A., J. McCrae, R. Cyganiak and A. Jentzsch, *The Linking Open Data cloud diagram*, 2017, http://lod-cloud.net/, accessed at Jul 2017.

53. Linked Data, *Connect Distributed Data across the Web*, 2017, http://linkeddata.org/, accessed at April 2017.

54. Dornescu, I. and C. Orăsan, "Densification: Semantic document analysis using Wikipedia", *Natural Language Engineering*, Vol. 20, pp. 469–500, 10 2014.

55. Gruetze, T., G. Kasneci, Z. Zuo and F. Naumann, "CohEEL: Coherent and efficient named entity linking through random walks", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 37, No. 0, 2016.

56. Jovanovic, J., E. Bagheri, J. Cuzzola, D. Gasevic, Z. Jeremic and R. Bashash, "Automated Semantic Tagging of Textual Content", *IT Professional*, Vol. 16, No. 6, pp. 38–46, Nov 2014.

57. Kiryakov, A., B. Popov, I. Terziev, D. Manov and D. Ognyanoff, "Semantic annotation, indexing, and retrieval", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 2, No. 1, pp. 49 – 79, 2004.

58. Rospocher, M., M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger and T. Bogaard, "Building event-centric knowledge graphs from news", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 37-38, pp. 132–151, 2016.

59. Müller, H.-M., E. E. Kenny and P. W. Sternberg, "Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature", *PLOS Biology*, Vol. 2, No. 11, 09 2004.

60. van Aggelen, A., L. Hollink, M. Kemman, M. Kleppe and H. Beunders, "The debates of the European Parliament as Linked Open Data", *Semantic Web Journal*, 2015.

61. Rocca, P. D., S. Senatore and V. Loia, "A semantic-grained perspective of latent knowledge modeling", *Information Fusion*, Vol. 36, pp. 52 – 67, 2017.

62. Kapanipathi, P., F. Orlandi, A. P. Sheth and A. Passant, "Personalized filtering of the Twitter stream", *Proceedings of the 10th International Semantic Web Conference*, October 2011.

63. Sahito, F., A. Latif and W. Slany, "Weaving Twitter stream into Linked Data a proof of concept framework", *7th International Conference on Emerging Technologies*, pp. 1–6, September 2011.

64. Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", *Journal of machine Learning research*, Vol. 3, No. Jan, pp. 993–1022, 2003.

65. Mehrotra, R., S. Sanner, W. Buntine and L. Xie, "Improving LDA Topic Models for Microblogs via Tweet Pooling and Automatic Labeling", *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pp. 889–892, ACM, New York, NY, USA, 2013.

66. Bauer, S., A. Noulas, D. O. Séaghdha, S. Clark and C. Mascolo, "Talking Places: Modelling and Analysing Linguistic Content in Foursquare", *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 348–357, Sept 2012.

67. Rosen-Zvi, M., T. Griffiths, M. Steyvers and P. Smyth, "The Author-topic Model for Authors and Documents", *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pp. 487–494, AUAI Press, Arlington, Virginia, United States, 2004.

68. Yan, X., J. Guo, S. Liu, X. Cheng and Y. Wang, "Learning Topics in Short Texts by Non-negative Matrix Factorization on Term Correlation Matrix", *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 749–757, 2013.

69. Lee, D. D. and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", *Nature*, Vol. 401, No. 6755, pp. 788–791, 1999.

70. Lehmann, J., B. Gonçalves, J. J. Ramasco and C. Cattuto, "Dynamical Classes of Collective Attention in Twitter", *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pp. 251–260, ACM, New York, NY, USA, 2012.

71. Miller, G. A., "WordNet: A Lexical Database for English.", *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41, 1995.

72. Harabagiu, S. and A. Hickl, "Relevance Modeling for Microblog Summarization", *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

73. Twitter, *Twitter Developer Documentation*, 2017, https://dev.twitter.com/rest/public, accessed at April 2017.

74. Wikipedia, the free encyclopedia, *Article titles*, 2017, http://en.wikipedia.org/wiki/Wikipedia:Article_titles, accessed at April 2017.

75. Salton, G. and C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information Processing & Management*, Vol. 24, No. 5, pp. 513–523, 1988.

76. TagMe, *TagMe API Documentation*, 2017, https://services.d4science.org/web/tagme/documentation, accessed at April 2017.

77. Schmachtenberg, M., C. Bizer and H. Paulheim, *State of the LOD Cloud 2014*, 2014, http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/, accessed at April 2017.

78. Suchanek, F. M., G. Kasneci and G. Weikum, "YAGO: A Large Ontology from Wikipedia and WordNet", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 3, pp. 203 – 217, 2008, world Wide Web Conference 2007Semantic Web Track.

79. Guha, R. V., D. Brickley and S. Macbeth, "Schema.Org: Evolution of Structured Data on the Web", *Commun. ACM*, Vol. 59, No. 2, pp. 44–51, Jan. 2016.

80. Brickley, D. and L. Miller, *FOAF vocabulary specification 0.91*, Tech. rep., Citeseer, 2007.

81. Wikidata, *Wikidata*, 2017, http://www.wikidata.org, accessed at April 2017.

82. Wikidata, *SPARQL endpoint*, 2017, https://query.wikidata.org/bigdata/namespace/wdq/sparql, accessed at April 2017.

83. DbPedia, *Virtuoso SPARQL Query Editor*, 2017, http://dbpedia.org/sparql, accessed at April 2017.

84. W3C Semantic Web Interest Group, *Basic Geo (WGS84 lat/long) Vocabulary*, 2017, https://www.w3.org/2003/01/geo/, accessed at April 2017.

85. GeoNames, *GeoNames Ontology - Geo Semantic Web*, 2017, http://www.geonames.org /ontology/, accessed at April 2017.

86. W3C, *Time Ontology URL*, 2017, https://www.w3.org/2006/time, accessed at April 2017.

87. W3C, *Time Ontology*, 2017, https://www.w3.org/TR/owl-time/, accessed at April 2017.

88. Duerst, M., W3C, M. Suignard and Microsoft Corporation, *Internationalized Resource Identifiers (IRIs)*, 2005, http://www.ietf.org/rfc/rfc3987.txt, accessed at April 2017.

89. Berners-Lee, T., W3C/MIT, R. Fielding, Day Software, L. Masinter and Adobe Systems, *Uniform Resource Identifier (URI): Generic Syntax*, 2005, http://www.ietf.org/rfc/rfc 3986.txt, accessed at April 2017.

90. W3C, *Namespaces*, 2017, https://www.w3.org/TR/rdf-sparql-query/\#docNamespac es, accessed at April 2017.

91. PostgreSQL, *PostgreSQL: The world's most advanced open source database*, 2017, https: //www.postgresql.org/, accessed at April 2017.

92. Apache, *Apache Solr*, 2017, http://lucene.apache.org/solr/, accessed at April 2017.

93. Apache Lucene, *Apache Lucene Core*, 2017, https://lucene.apache.org/core/, accessed at April 2017.

94. Apache Jena, *Apache Jena Fuseki*, 2017, https://jena.apache.org/documentation/fuse ki2/, accessed at April 2017.

95. Bailey, F., *Phirehose*, https://github.com/fennb/phirehose, accessed at April 2017.

96. Yıldırım, A., S. Üsküdarlı and A. Özgür, "Identifying Topics in Microblogs Using Wikipedia", *Plos One*, Vol. 11, No. 3, pp. 1–20, 03 2016.

97. Twitter, *Twitter Developers, GET statuses/sample*, 2017, https://dev.twitter.com/streaming/reference/get/statuses/sample, accessed at April 2017.

98. WordNet, *WN(1WN) manual page*, 2017, https://wordnet.princeton.edu/wordnet/man/wn.1WN.html, accessed at May 2017.

99. BounTI, *Stopwords*, 2017, http://soslab.cmpe.boun.edu.tr/bounti/stopwords.txt, accessed at May 2017.

100. Wikipedia, *Size Comparisons*, 2017, https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons, accessed at April 2017.

101. Sharp, A., *Dispatch from the Denver debate*, https://blog.twitter.com/2012/the-final-2012-presidential-debate, accessed at Oct 2012.

102. Sharp, A., *Recapping the VP debate*, https://blog.twitter.com/2012/recapping-the-vp-debate, accessed at Oct 2012.

103. Sharp, A., *Twitter at the Town Hall Debate*, https://blog.twitter.com/2012/twitter-at-the-town-hall-debate, accessed at Oct 2012.

104. Sharp, A., *The Final 2012 Presidential Debate*, https://blog.twitter.com/2012/the-final-2012-presidential-debate, accessed at Oct 2012.

105. Twitter, *Twitter Developers, POST statuses/filter*, 2017, https://dev.twitter.com/streaming/reference/post/statuses/filter, accessed at April 2017.

106. Hripcsak, G. and A. S. Rothschild, "Agreement, the f-measure, and reliability in information retrieval", *Journal of the American Medical Informatics Association*, Vol. 12, No. 3, pp. 296–298, 2005.

107. New York Times, *The First Presidential Debate*, 2017, http://www.nytimes.com/interactive/2012/10/04/us/politics/20120804-denver-presidential-debate-obama-romney.html, accessed at October 2012.

108. The CNN Political Unit, *Transcript of Wednesday's presidential debate*, 2017, http://edition.cnn.com/2012/10/03/politics/debate-transcript/, accessed at Oct 2012.

109. DBPedia, *DBpedia 3.0 Downloads*, 2017, http://oldwiki.dbpedia.org/Downloads30\#extendedabstracts, accessed at May 2017.

110. Little, C. and S. Cox, *Time Ontology in OWL*, W3C working draft, W3C, Jul 2016, https://www.w3.org/TR/2016/WD-owl-time-20160712/.

111. Boğaziçi University, *SBounTi topic SPARQL interface*, 2017, http://193.140.196.97:3030/dataset.html, accessed at April 2017.

112. Open Calais, *Thomson Reuters*, 2017, http://www.opencalais.com/, accessed at April 2017.

113. Foursquare, *Foursquare*, 2017, https://tr.foursquare.com/, accessed at May 2017.

114. DBpedia, *DBpedia query that retrieves all entities of type foaf:Organization*, 2017, https://dbpedia.org/sparql?query=select+\%3Fentity+where+\%7B\%0D\%0A\%3Fentity+rdf\%3Atype+foaf\%3AOrganization\%0D\%0A\%7D, accessed at May 2017.

115. Suchanek, F. M., G. Kasneci and G. Weikum, "Yago: A Core of Semantic Knowledge", *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pp. 697–706, ACM, New York, NY, USA, 2007.

116. Steiner, T., R. Verborgh, R. Troncy, J. Gabarro and R. Van De Walle, "Adding Realtime Coverage to the Google Knowledge Graph", *Proceedings of the 2012th International Conference on Posters & Demonstrations Track - Volume 914*, ISWC-PD'12, pp. 65–68, CEUR-WS.org, Aachen, Germany, Germany, 2012.

117. DBpedia, *DBpedia query that retrieves all entities of type foaf:Group*, 2017, https://dbpedia.org/sparql?query=select+\%3Fentity+where+\%7B\%0D\%0A \%3Fentity+rdf\%3Atype+foaf\%3AGroup\%0D\%0A\%7D, accessed at May 2017.

118. Tonon, A., M. Catasta, R. Prokofyev, G. Demartini, K. Aberer and P. Cudré-Mauroux, "Contextualized ranking of entity types based on knowledge graphs", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 37-38, pp. 170–183, 2016.

119. Swartz, A., "MusicBrainz: a semantic Web service", *IEEE Intelligent Systems*, Vol. 17, No. 1, pp. 76–77, Jan 2002.

120. Wilton, P., J. Tarling and J. McGinnis, *Stroyline Ontology*, 2017, http://www.bbc.co.uk/ontologies/storyline, accessed at May 2013.

121. Barbieri, D. F., D. Braga, S. Ceri, E. Della Valle and M. Grossniklaus, "C-SPARQL: SPARQL for Continuous Querying", *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pp. 1061–1062, ACM, New York, NY, USA, 2009.

122. Virtuoso, *OpenLink Virtuoso Home Page*, 2017, https://virtuoso.openlinksw.com/, accessed at May 2017.

123. Qiu, M., *Latent Dirichlet Allocation (LDA) model for Microblogs (Twitter, weibo etc.)*, 2017, https://github.com/minghui/Twitter-LDA, accessed at Jan 2012.

124. Aiello, L. M., G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris and A. Jaimes, "Sensing Trending Topics in Twitter", *IEEE Transactions on Multimedia*, Vol. 15, No. 6, pp. 1268–1282, Oct 2013.

125. Yilmaz, E., J. A. Aslam and S. Robertson, "A New Rank Correlation Coefficient for Information Retrieval", *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pp. 587–594, ACM, New York, NY, USA, 2008.

126. Bar-Ilan, J., M. Mat-Hassan and M. Levene, "Methods for comparing rankings of search engine results", *Computer Networks*, Vol. 50, No. 10, pp. 1448–1463, 2006, i. Web Dynamics II. Algorithms for Distributed Systems.

APPENDIX A: ADDITIONAL STOPWORDS FOR TWITTER POST-PROCESSING IN BOUN-TI

Table A.1. Stopwords used in processing tweets. In addition to the items in stopwords lists that is often used in text processing tasks, the items in this list are considered in Boun-TI prototype. See [99] for the full list of stopwords.

Stopword
ain
wasn
d
ve
isn
ll
rt
doesnt
amp
amp;
gain
follow
follower
followers
love
good
retweet
retweets
Continued on next page

Table A.1. Stopwords used in processing tweets. In addition to the items in stopwords lists that is often used in text processing tasks, the items in this list are considered in BOUN-TI prototype. See [99] for the full list of stopwords. (cont.)

Stopword
tweet
twitter
don
unfollowers
unfollow
lol
today
http
https
retweet
rts
happy
person
people
didn
followback
day
days

APPENDIX B: SBOUN-TI TEMPORAL EXPRESSION IDENTIFIERS

Table B.1. The enitities linked to temporal expression spots.

Spot	**Entity**
tdy	topico:Today
today	topico:Today
tonight	topico:Tonight
night	topico:Night
now	topico:Now
tomorrow	topico:Tomorrow
tmrw	topico:Tomorrow
tmrrw	topico:Tomorrow
yesterday	topico:Yesterday
ystrdy	topico:Yesterday
january	topico:January
february	topico:February
march	topico:March
april	topico:April
may	topico:May
june	topico:June
july	topico:July
august	topico:August
september	topico:September
october	topico:October
Continued on next page	

Table B.1. The enitities linked to temporal expression spots. (cont.)

Spot	**Entity**
november	topico:November
december	topico:December
spring	topico:Spring
summer	topico:Summer
winter	topico:Winter
fall	topico:Fall
evening	topico:Evening
morning	topico:Morning
this evening	topico:ThisEvening
this morning	topico:ThisMorning
this afternoon	topico:ThisAfternoon
last year	topico:LastYear
last week	topico:LastWeek
last month	topico:LastMonth
last night	topico:LastNight
sunday	topico:Sunday
monday	topico:Monday
tuesday	topico:Tuesday
wednesday	topico:Wednesday
thursday	topico:Thursday
friday	topico:Friday
saturday	topico:Saturday

APPENDIX C: ENTITY LINK IMPROVEMENTS

Table C.1. Some spots and their corresponding initial and relinked entities

spot	**Links before improvement**	**Links after improvement**
donald	dbr:Donald_Nixon	dbr:Donald_Trump
donald	dbr:Peter_Donald	dbr:Donald_Trump
donald	dbr:Donald_Regan	dbr:Donald_Trump
donald	dbr:Howard_Donald	dbr:Donald_Trump
hillary	dbr:Edmund_Hillary	dbr:Hillary_Clinton
hillary	dbr:Richard_Hillary	dbr:Hillary_Clinton
trump	dbr:Trump_(magazine)	dbr:Donald_Trump
trump	dbr:First_Trump	dbr:Donald_Trump
trump	dbr:Trump	dbr:Donald_Trump

APPENDIX D: RESULT OF QUERIES IN FIGURE 6.14

Table D.1. Result of queries in Figure 6.14 which are the politicians existing in topics.

Politician
dbr:Abraham_Lincoln
dbr:Bill_Clinton
dbr:Colin_Powell
dbr:Richard_Nixon
dbr:Ronald_Reagan
dbr:Bernie_Sanders
dbr:Kelly_Ayotte
dbr:Tim_Kaine
dbr:John_Kerry
dbr:George_W._Bush
dbr:Hosni_Mubarak
dbr:Antonio_Gramsci
dbr:Saddam_Hussein
dbr:Vladimir_Putin
dbr:Donald_Trump
dbr:George_Wallace
dbr:Jesse_Jackson
dbr:Tim_Scott
dbr:Gary_Johnson
dbr:Hillary_Clinton
dbr:Al_Gore
dbr:Gerald_Ford
dbr:Jack_Dalrymple
dbr:Heidi_Heitkamp
dbr:Recep_Tayyip_Erdoğan