PROBABILISTIC MODELS FOR CONTROVERSY SEARCH WITH CONTENTION, LANGUAGE, AND TIME

A Thesis Presented

by

MYUNGHA JANG

Submitted to the Graduate School of the University of Massachusetts Amherst in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Dec 2018

College of Information and Computer Sciences
PROBABILISTIC MODELS FOR CONTROVERSY SEARCH WITH CONTENTION, LANGUAGE, AND TIME

A Thesis Presented
by
MYUNGHA JANG

Approved as to style and content by:

________________________________________
James Allan, Chair

________________________________________
W. Bruce Croft, Member

________________________________________
Brendan O’Connor, Member

________________________________________
Weiai Wayne Xu, Member

James Allan, Chair of the Faculty
College of Information and Computer Sciences
ABSTRACT

PROBABILISTIC MODELS FOR CONTROVERSY SEARCH WITH CONTENTION, LANGUAGE, AND TIME

DEC 2018

MYUNGHA JANG

B.Sc., EWHA WOMANS UNIVERSITY

M.Sc., POSTECH

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Navigating controversial topics on the Web encourages social awareness, supports civil discourse, and promotes critical literacy. Because controversy search tasks can often be overwhelming for users, there have been substantial efforts to automate this process. While existing approaches have worked well in practice, they are narrow in scope and exhibit limited performance. Addressing these drawbacks, we propose a probabilistic approach to controversy search tasks. Specifically, we present a set of methods that address how to identify whether the topic of a document is controversial and how to explain why the topic is controversial. To understand the theoretical grounding of the state-of-the-art algorithm, we first derive an underlying probabilistic model that explains the state-of-the-art controversy detection algorithm. From the model, we identify shortcomings of the current approach. We propose a modified framework to address the drawbacks of the algorithm. We also revisit and challenge
the two properties that the state-of-the-art model depends upon and propose a new probabilistic model that considers controversial language. We show that this framework substantially outperforms the start-of-the-art algorithm. We point out that the current approaches for controversy detection do not consider time while controversy is a dynamically changing phenomenon. This causes current methods to have delays in recognizing emerging controversial topics or exaggerated effects on outdated controversies. For the proposed work, we will address time-adaptable controversy detection by estimating the controversy trend of topics beyond the observed conflicts. Finally, we offer a method that explains controversy by generating a summary of each stance. This method ranks social media postings using a score of how likely the given post can be a representative summary of controversy.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xi</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Contributions and Anticipated Contributions</td>
<td>4</td>
</tr>
<tr>
<td>2. RELATED WORK</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Controversy Detection on the Web</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1 Identifying Controversy in Webpages and News Articles</td>
<td>8</td>
</tr>
<tr>
<td>2.1.2 Measuring the level of Controversy in Wikipedia</td>
<td>9</td>
</tr>
<tr>
<td>2.1.3 Recognizing and Measuring Controversy on Social Media</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Stance summarization on Social Media</td>
<td>10</td>
</tr>
<tr>
<td>2.2.1 Twitter Summarization</td>
<td>10</td>
</tr>
<tr>
<td>2.2.2 Stance Detection on Twitter</td>
<td>11</td>
</tr>
<tr>
<td>3. PROBABILISTIC MODELING OF CONTROVERSY DETECTION USING CONTENTION</td>
<td>12</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Background: Theoretical Models to Define Controversy</td>
<td>14</td>
</tr>
<tr>
<td>3.3 A Probabilistic Model of ( knn-WC ) Algorithm</td>
<td>15</td>
</tr>
<tr>
<td>3.3.1 Estimating ( P(c</td>
<td>w) ) using Contention</td>
</tr>
<tr>
<td>3.4 Discussion</td>
<td>21</td>
</tr>
</tbody>
</table>
## 4. REVISITING AND IMPROVING WIKIPEDIA-BASED CONTROVERSY DETECTION

### 4.1 Revisiting the assumptions of the algorithm

- **4.1.1 The Limitation of a Single Document Query**
- **4.1.2 The Limitation of Wikipedia Controversy Features**

### 4.2 Proposed Framework

- **4.2.1 TileQuery Generation**
  - **4.2.1.1 Document Segmentation**
  - **4.2.1.2 Query Generation**
  - **4.2.1.3 Aggregating the Ranked Lists**
- **4.2.2 Smoothing Controversy score of Wikipages**
  - **4.2.2.1 Constructing Wikipages’ Graph**
  - **4.2.2.2 Graph-based Smoothing**
- **4.2.3 Aggregation and Voting**

### 4.3 Experiments

- **4.3.1 Dataset**
- **4.3.2 Results and Discussion**

### 4.4 Conclusion

## 5. CONTROVERSY LANGUAGE MODELS

### 5.1 Introduction

### 5.2 Proposed Model

### 5.3 Evaluation

### 5.4 Results

### 5.5 Proposed Work: Investigating Generalizability

### 5.6 Proposed Work: Investigating Explainability of the Model

### 5.7 Conclusion

## 6. PROPOSED RESEARCH: TIME-ADAPTABLE CONTROVERSY DETECTION

### 6.1 Introduction

- **6.1.1 The Dynamic Nature of Controversy**
- **6.1.2 Monotonicity of Controversy Scores in Wikipedia**
6.1.3 Proposed Work: Time-window based Baselines .............. 46
6.1.4 What happens between the controversial peaks? ............ 48

6.2 A Sketch of the Proposed Method ................................ 48
6.3 Evaluation ..................................................... 49

7. EXPLAINING CONTROVERSY ON SOCIAL MEDIA .......... 50

7.1 Introduction .................................................. 50
7.2 Approach ..................................................... 51
7.2.1 Ranking Model ............................................ 52
7.2.2 Estimating Stance-indication............................... 53
7.2.2.1 Identifying Stance Hashtags (\(H_A, H_B\)) ............ 53
7.2.2.2 Estimating \(P(h|\tau)\) via Latent Hashtags ............... 54
7.2.3 Estimating the level of articulation ......................... 55
7.2.4 Summary Selection ........................................ 56

7.3 Evaluation ..................................................... 56
7.3.1 Experiment Setup ......................................... 56
7.3.2 Results and Discussion .................................... 57

7.4 Conclusion and Future Work ..................................... 60
7.5 Proposed Work: More Rigorous Evaluation ..................... 60

8. RESEARCH PLAN .................................................. 61

BIBLIOGRAPHY ...................................................... 62
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>A summary of notations used in our probabilistic framework</td>
</tr>
<tr>
<td>4.1</td>
<td>An example of M score and C score for Wikipages on “Abortion” that most sub-pages on “Abortion” have controversy scores close to 0.</td>
</tr>
<tr>
<td>4.2</td>
<td>An example of two controversy scores on several Wikipages on “Abortion”, before and after score smoothing</td>
</tr>
<tr>
<td>4.3</td>
<td>Accuracy, F1, and the best parameters in 5-fold runs for different query and inferred score settings.</td>
</tr>
<tr>
<td>4.4</td>
<td>Improvements of accuracy and F1 score between runs (bold: statistically significant)</td>
</tr>
<tr>
<td>5.1</td>
<td>The notation summary of controversy language model</td>
</tr>
<tr>
<td>5.2</td>
<td>An example of controversy-indicative terms.</td>
</tr>
<tr>
<td>5.3</td>
<td>The accuracy of the models. $k$NN-WC+ refers to the improved version of the algorithm presented in Chapter 4.</td>
</tr>
<tr>
<td>5.4</td>
<td>Wikipedia-Based Controversy Detection Approaches. All Controversy Language Model (CLM) approaches have significant improvements over their respective $k$NN-WC counterpart at the $p &lt; 0.05$ level.</td>
</tr>
<tr>
<td>5.5</td>
<td>Language Models built from documents relevant to Cramer’s controversial terms (Cramer, 2011). Collection size $</td>
</tr>
</tbody>
</table>
5.6 Language Models built from Cramer’s terms and existing lexicons on DBPedia. We find that “controversy” is the most indicative term, and that “saga” is no better than random. Combining terms led to no improvement over “controversy” alone. “TF10” indicates that the TF10 query is used to represent a document whereas “Full” indicates that the full text of the document is used as a query.

5.7 A comparison of lexicons built manually and through crowd-sourcing in prior work to our automatically derived language models A (*) indicates significant improvement over the best lexicon approach.

7.1 An example of good (top) and bad (bottom) summary tweets on “Abortion” posted on Nov 4, 2016. The good summaries are selected from our method. Examples of stance hashtags are marked in bold.

7.2 The features used to train a regression model for predicting the level of tweet articulation.

7.3 The amount of data used to train Tweet2Vec and summary generation. The number in parentheses refers to the number of tweets published by the stance community.
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 A simple Bayesian network $k$NN-WC model is based on. D: Document, T: Topic, and C: Controversiality.</td>
<td>16</td>
</tr>
<tr>
<td>4.1 An example of the constructed graph for <em>Abortion</em> and two different sub-graphs selected based on the two methods.</td>
<td>27</td>
</tr>
<tr>
<td>4.2 A graph constructed around the query “Same-sex marriage in Sweden”. The scores are borrowed from three neighbors that have more revision counts than the query page by weighted smoothing.</td>
<td>29</td>
</tr>
<tr>
<td>6.1 Controversy computed by P score (Jang et al., 2017) among all daily tweets by date for The Dress (left), Brexit (center) and 2016 US Elections (right), reported among those Gardenhose tweets with an explicit stance. Notable peaks are annotated with associated events around that time. All dates are in UTC.</td>
<td>44</td>
</tr>
<tr>
<td>6.2 “Time evolution of the controversy measure of the article about Michael Jackson. A: Jackson is acquitted on all counts after five month trial. B: Jackson makes his first public appearance since the trial to accept eight records from the Guinness World Records in London, including Most Successful Entertainer of All Time. C: Jackson issues Thriller 25. D: Jackson dies in Los Angeles.” Source: <a href="http://wwm.phy.bme.hu/">http://wwm.phy.bme.hu/</a></td>
<td>46</td>
</tr>
<tr>
<td>6.3 An estimated trend line of controversy (dotted) over the observed trend (solid).</td>
<td>47</td>
</tr>
<tr>
<td>7.1 The evaluation results by the methods. The rightmost four bars are our methods.</td>
<td>58</td>
</tr>
<tr>
<td>7.2 The user study results by the topics. The rightmost four bars in each topic are our methods. We did not include SumBasic in the graph because it was the worst method for all topics, being preferred only 8% of times overall.</td>
<td>58</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

*Controversy search*, which refers to online user activities that seek information on controversial topics, has become increasingly more important in a wide range of informational retrieval tasks. For example, users might search for medical topics and face challenges with misinformation about fraudulent treatments or spurious links between vaccines and autism. In the political sphere, users might search for presidential candidates to learn about their campaigns or last night’s presidential debate to make up their mind whom to vote for, or search for a new tax bill draft that has been controversial. Some users just heard about a data scandal between Cambridge Analytica and Facebook (Granville, 2018)\(^1\), and would like to understand the controversy and how it might affect them.

However, growing polarization on controversial topics has resulted in the spread of “Filter Bubbles” (Pariser, 2011), which makes it difficult for users to discover diverse opinions on them. As high-stake information on controversial topics may lead to real-life decisions and actions, effectively assisting controversy search has lasting impact on the society. Therefore, controversy search on the Web therefore not only encourages social awareness, supports civil discourse, and promotes critical literacy but also enables users to make more informed decisions.

As online sources become the primary source for information, the need for effective controversy search has grown substantially. According to Pew Research Center’s

\(^1\)The Facebook – Cambridge Analytica data scandal involves a controversy over the collection of personal information of up to 87 million Facebook users that has been improperly obtained by a political consulting firm Cambridge Analytica.
survey on how Americans get their news (2016), online platforms, including news websites, social media, and mobile apps, were one of the most dominant sources of information. Although television has been the most used platform among older generations, the general trend suggests that online platforms will soon replace many traditional sources of information. Because online content has gradually replaced television and print news, people are now exposed to an overwhelming amount of information that may be untrustworthy and biased. This suggests that the role of controversy search has become even more crucial for consuming controversial topics from online sources.

Controversy search systems aim at helping users to understand the controversial aspects of the topic that they search for. The goal of the system includes not only helping users who actively seek to understand a certain controversial topic, but also alerting users who were not aware that this topic that they are reading is controversial. This is because unsuspecting users are likely to be misled by biased content, especially if they not aware that the topic is controversial. Therefore, the system overall aims to modulate the search results for controversial topics by systematically mediating the bias and the filter bubble phenomenon.

Building controversy search systems, however, can be challenging because controversy search is a complex task for a few reasons. One reason is that determining the extent of role of the controversy search system is a complicated issue. Dori-Hacohen et al. (2015) brought up two open questions that need to be considered regarding the role of the search system. First, how much should the system should help users explicitly in finding content of different stances is? For example, should the system only show the results that match the keywords of the user queries even if the result contains the biased results, or make users aware that there are other stances if the query involves controversial topics? Second, should the system deliver every result available that are ungrounded, fraudulent, and even harmful? For example, should
the system still present a document of “Issel treatment” as a result for “cancer treatment” when the document contains the query while if the system knows that it is also listed as a “dubious treatment” by QUACKWATCH.COM²?

In addition to these ethical aspects, controversy search bears numerous technical challenges. While the sub-tasks have a different set of specific challenges, the commonly-shared challenge is that there is a multitude of subtleties in information of controversial topics. For example, while some topics might have a single correct answer, others, especially those that require moral judgment, have several possible answers. The same topic can be controversial to those who care more and know more details about it, while it is not controversial to those who either don’t care or don’t know much about it (Jang et al., 2017). For these reasons, it is even challenging to computationally define controversy, hence causing other related tasks (e.g., recognizing controversy, explaining controversy, etc) inherently difficult.

Unfortunately, prevailing techniques in information retrieval, which are typically designed for retrieving relevant information, are not optimized for controversy search. For example, existing search engines are unlikely to reveal controversial topics to users unless they already know about them (Gerhart, 2004). There is a higher call for search engines to detect these queries and address them appropriately (Dori-Hacohen et al., 2015). Earlier work presented an algorithm for classifying controversy in Web documents (Dori-Hacohen and Allan, 2015; Jang and Allan, 2016). Social media is also increasingly a place where controversy discourse is being shaped and dynamically evolves. However, we currently lack a tool for effectively navigating the postings around controversy in social media. For example, users have to manually examine postings to find the arguments of conflicting stances that make up the controversy.

²QuickWatch is a website that allows people report health-related frauds, myths, or any quackery-related information in medicine.
Towards the goal of building a system that supports controversy search, we investigate approaches to handle two types of questions: (1) “Does this document discuss a controversial topic?” and (2) “Why is this topic controversial?” While the second task is novel as we propose, the first task has been handled via techniques that classify a document whether it discusses a controversial topic. There have been several algorithms that have been targeted for this task (Dori-Hacohen and Allan, 2013; Dori-Hacohen and Allan, 2015; Beelen et al., 2017; Jang and Allan, 2016), however, little work has explored this problem from a modeling perspective. Therefore, gaps still remain in our theoretical and practical understanding. In this thesis, we study probabilistic models that address the above two questions but that also have an explanatory power in them.

1.1 Contributions and Anticipated Contributions

Modeling Controversy Detection in the Web:

- Deriving a probabilistic framework for controversy detection on Web (Completed): To understand the model behind the prevailing algorithms for controversy detection, we analyze the state-of-the-art algorithm ($k$NN-WC) and derive the underlying model that explains the theoretic ground of algorithm (Dori-Hacohen and Allan, 2015). We identify a few assumptions and properties that the model holds. In the subsequent chapters, we challenge them to analyze and improve the state-of-the-art algorithm, and propose a new model that holds complementary properties.

- Improving the state-of-the-art controversy detection algorithm (completed): From the derived model, we identify two drawbacks of the $k$nn-WC algorithm and propose an improved version of the algorithm with these issues addressed. Our derived model explains that the success of the algorithm depends on the
correctness of the following two assumptions: (1) A query generated from the document retrieves Wikipages that represent the document’s topics, and (2) Wikipages that discuss controversial topics will show a high level of contention among the editors of the page. We show that these assumptions are inaccurate in the way that current algorithm estimates the probabilistic components. Therefore we suggest solutions based on two findings: First, generating multiple queries from several semantically-coherent paragraphs is more effective in finding relevant Wikipedia topics. Second, since a controversial discussion that contribute to a controversy score usually takes place in a few representative pages among Wikipedia pages of similar controversial topics, inferring the controversy score from taxonomically-related Wikipedia pages makes the controversy score more accurate. The new algorithm that combined the two fixes significantly improves the controversy detection task in Webpages by 6% (Jang and Allan, 2016).

• Presenting a new model of contention and language (completed): Motivated by the drawbacks of knn-WC model, we argue that contention alone is neither a sufficient nor reliable feature for detecting controversy as contention is a feature that is observed sparingly. We introduce a new model, Controversy Language Model (CLM), where the “contention” feature was transformed to a “language” feature by building a language model from contentious topics. Our CLM classifies the document as controversial when its language is more similar to that of controversial topics than to that of general topics and the opposite for the non-controversial case. To evaluate its efficacy, we experiment with various ways of constructing controversial topics. We show that a CLM that is built with Wikipedia articles that contain several controversy-related keywords was 14% more effective in AUC in identifying controversial Webpages in our dataset, sig-
significantly outperforming the original algorithm that only relied on contention features. (Jang et al., 2016)

- Investigating generalizability and explainability of CLM (anticipated): For the proposed future work, we will further examine the utility of CLM from two aspects, generalizability and explainability. The two specific research questions we aim to answer here are (1) How well does CLM generalize on predicting controversy for emerging topics? and (2) How well do the top keywords in the document that CLM used to classify it as controversial effectively explain why the document is controversial to users?. To answer the first question, we might curate a new dataset that allows us evaluate the prediction of emerging topics.

- Time-adaptable Controversy Detection (anticipated): To correctly estimate the level of controversy that constantly evolves over time, we propose a new controversy function that takes a time of query into consideration. We first investigate a straightforward solution of computing the automated controversy scores by only considering the signals occurred for a window of given time. We will propose a technique that estimates the controversy trend line by using the controversial peaks as reliable sample points of the real trend. We define a concept of controversy decay model as a model that reflects that the level of controversy naturally decays over time. Our technical contribution includes finding a proper model that simulates the controversy decay phenomenon and estimating a topic-specific decay speed. We plan to evaluate this work by combining it to CLM approach for controversy detection. As our currently available annotated dataset does not have any time information, we will to create a new dataset including articles on the same topic but from different times over the past 10 years.

Modeling Stance Summarization:
Hashtag-based Stance Summarization (completed): We pose a novel problem of explaining controversy on Twitter via generating a summary of two conflicting stances that make up controversy. We first characterize a few aspects that a desirable summary should satisfy, namely: Stance-indication, Articulation, and Topic-relevance. This model defines that a good summary should include tweets whose stance is clearly indicated, whose language is articulate, and the content is relevant to the given controversial topic. Specifically, we use the Twitter’s retweet network property to first find user stance communities, and extract the stance hashtags that are distinctively used in each community. We train tweet embedding using hashtags as labels to obtain the probability that tweets are likely to generate a given hashtag for all hashtags. We show that tweets that have semantically close text to the top stance hashtags that best describe the stance community while being articulate and relevant to the topic are more likely to be an effective summary. Our human evaluation shows that our summaries are more preferred over other baseline summaries (Jang and Allan, 2018).

Intrinsic controversy summary evaluation (anticipated): We previously evaluated our results by surveying people on Amazon Mechanical Turk. However, this evaluation method is not scalable because it is expensive and slow. In addition, simply by asking the overall quality of the summary of the results, it is hard to obtain a deeper analysis on the quality of the summaries in depth, such as in which aspect one summary was better than the other. Hence, we will investigate an intrinsic evaluation method that is similar to traditional recall and precision in a controversy summary context. This will lay a solid path for future endeavors towards the problem, while allowing us to further understand the nature of the problem by studying the aspects that a controversy summary should satisfy.
CHAPTER 2
RELATED WORK

Our work is related to several areas, and builds on the findings of the previous research. We categorize the areas largely by controversy detection on the Web and stance summarization on social media, and discuss the various related sub-problems and efforts that have been studied by our community to address them. Some of the directly related work are discussed more in depth in the relevant chapter.

2.1 Controversy Detection on the Web

Depending on from which medium the controversy is being detected, the detection tasks poses different challenges and solutions. The two most-studied mediums are Wikipedia and social media (e.g., Twitter), where user-interaction feature has been recognized as one of the most important features to detect the controversy from (Kittur et al., 2007; Yasseri et al., 2012; Garimella et al., 2016; Dori-Hacohen, 2017). The other type of sources include general web-pages and news articles, which usually do not have any meta-data such as user-interaction features, but just the text of the documents. While our work focuses on detecting controversy on webpages, we discuss the work in these other sub-areas as well.

2.1.1 Identifying Controversy in Webpages and News Articles

To the best of our knowledge, Dori-Hacohen and Allan's work (2015) was the first attempt to extend the controversy detection problem to general web-pages in open-domain. Their work builds upon the small but growing body of work that investigates
controversy detection on Wikipedia (Jankowski-Lorek et al., 2014). They begin by generating a query from a web page, and retrieving the $K$ nearest neighbors from Wikipedia. They create a binary classifier by aggregating controversy features that are computed in relevant Wikipedia pages (Yasseri et al., 2012; Das et al., 2013a).

While some past work uses sentiment as a signal when researching controversy (Cartright et al., 2009; Choi et al., 2010), others have argued that opinion and controversy are distinct and non-overlapping concepts (Awadallah et al., 2012). Mejova et al. (2014) argue that controversy and sentiment are not directly related. This is also supported by an experiment by Dori-Hacohen and Allan (2013) that using sentiment for controversy detection performs poorly on webpages.

Choi et al.’s work is a pioneering effort to identify controversy and controversial subtopics from news articles using various features, particularly a mixture model of topic and sentiment (Choi et al., 2010). Later, Beelen et al. (2017) also studied identifying controversy from news articles by investigating extensive features that indicate controversy from the document text as well as people’s comments. They showed that their comment-based method that considers the meta-data of comments of the news articles was more effective than a content-based approach that considers the text of the news articles for controversy detection in news articles.

There also have been a few attempts to detect controversial content with lexicons. Roitman et al. (2014) retrieve Wikipedia articles that contain as many relevant claims about controversial query topics using manually-curated controversy lexicon, and Mejova et al. (2014) use crowdsourcing to label controversial words.

2.1.2 Measuring the level of Controversy in Wikipedia

As Wikipedia contains manually tagged controversial articles by editors, machine-learning based methods approaches were trained to learn them. To estimate the level of controversy in Wikipages, information extracted from the edit-history, such as
revision count, number of unique editors, number of reverts, the number of editors participating in the edit-war, and their reputations have been exploited (Kittur et al., 2007; Yasseri et al., 2012). Sepehri Rad and Barbosa (2012) surveyed five established controversy detection algorithms on Wikipedia and compared their performances. Since they use data-source-specific features such as Wikipedia’s edit history features or Twitter’s social graph information, existing work cannot be easily generalized to controversy detection on arbitrary web-pages.

2.1.3 Recognizing and Measuring Controversy on Social Media

Existing work also has focused on identifying controversy on Twitter (Popescu and Pennacchiotti, 2010; Garimella et al., 2016; Fraisier et al., 2017). Garimella et al. and Fraisier et al. analyze user retweet or follow graphs, which signifies the formation of exclusive communities of like-minded people for controversial topics.

2.2 Stance summarization on Social Media

2.2.1 Twitter Summarization

There has been much work on summarizing Twitter postings while most of them focuses on summarizing events (Sharifi et al., 2010; Duan et al., 2012; Chakrabarti and Punera, 2011; Inouye and Kalita, 2011; Yulianti et al., 2016). Inouye et al. (Nenkova and Vanderwende, 2005) compare multiple summarization algorithms for Tweet data, and their extensive experiments suggest that the SumBasic algorithm produced the best F1-result in human evaluation, which we also adopt as a summarization baseline in this paper.

Some work has focused on generating contrastive summaries from opinionated text (Paul et al., 2010; Guo et al., 2015). Particularly, Guo et al. studied tweet data to find a controversy summary. They find a pair of contrastive opinions by integrating manually-curated expert opinions and clustering the pairs to generate a summary.
However, their model needs curated expert opinions, which requires constant human effort to maintain as the topic evolves.

2.2.2 Stance Detection on Twitter

Stance classification on Twitter has two main tasks: (1) classifying the text’s stance (against, favor, or neutral) given a topic, or (2) classifying the twitter users’ stances. The former task drew attention when 2016-SemEval Task 6 released a dataset of tweets with stance annotations (Mohammad et al., 2016a). The results of various approaches were shared after the competition (Mohammad et al., 2016b), and later more successful approaches were proposed including one that uses a bi-directional conditional LSTM for classifying the stance and opinion target on Twitter (Augenstein et al., 2016). For the latter type of task, Johnson and Goldwasser developed a method to classify stances of politicians on Twitter using relational representation (Johnson and Goldwasser, 2016). While stance detection is closely related to our problem, our goal is not to accurately classify the stances of all tweets. Our problem is also more robust to misclassification errors of stances as we take the tweets with highest stance confidence as part of the summary.
CHAPTER 3

PROBABILITY MODELING OF CONTROVERSY DETECTION USING CONTENTION

3.1 Introduction

This chapter discusses a probabilistic framework for the task of detecting controversy of a given web document. Dori-Hacohen and Allan (2013) first introduced the problem of detecting controversial topics in Web documents. The goal of this task is to make a binary classification on whether or not a given document discusses controversial topics. Dori-Hacohen and Allan conducted a pilot study to demonstrate that related Wikipedia pages can be used to detect controversy in web documents. They first mapped each query webpage to $k$ related Wikipedia pages (Wikipages), and used the annotated controversy level of the selected Wikipages to produce a final controversy score for the document. Later, they proposed the first fully-automated technique for this task (2015), which we call “$k$NN-WC algorithm”.

The $k$NN-WC algorithm has been shown to be effective. However, its lack of theoretical underpinning leaves a gap between our theoretical and empirical understanding in this problem. Why do we need a model when we already have an algorithm that works reasonably well? When an algorithm is instantiated from a theoretically-grounded model, we can obtain a better intuition of why the algorithm works. Having a model allows us to understand mathematical foundations and to evaluate a set of assumptions made to design the algorithm. This can help us to challenge the existing assumptions and develop better algorithms.

We know of only two efforts to examine a theoretic model for controversy (Dori-Hacohen, 2017; Kazimierz Zielinski and Jatowt, 2018). Because both of the proposed
models computationally define controversy as the disputes within a given community (or a ‘population’), they require auxiliary signals of disputes to estimate controversy, such as Wikipedia’s edit history or user interaction behaviors on social media. Therefore, those models are not directly applicable to Webpages that do not have any external signal but just text.

Although Dori-Hacohen and Allan leave the theoretical groundwork of the kNN-WC algorithm largely unexamined, we propose that the algorithm has been implicitly instantiated from an underlying model. We therefore analyze and derive a model for the kNN-WC algorithm. Our goal here is not to design a new model but instead to derive a model that explains the kNN-WC algorithm. We later demonstrate that deriving such a model can be used to extend the approach and design models and algorithms with substantially improved efficiency, accuracy, and generalizability. Specifically, deriving this model is a crucial step towards understanding this problem in many ways because it allows us to answer the following research questions:

- **Theoretical Understanding of the Problem**: What is the mathematical background of the model and what assumptions were made in the model?

- **Revisiting the algorithm**: How reasonable were the assumptions of the model? How accurately do the heuristics adopted by the algorithm estimate certain probabilities? Are there better ways to estimate them?

- **Testing a new hypothesis**: Once we understand the assumptions of the current model, we can challenge those assumptions and test a new hypothesis.

In this chapter, we will present a probabilistic model of kNN-WC. We show that kNN-WC model is based on a population-based controversy model (Dori-Hacohen, 2017), which assumes that “contention” within a given population is a primary measure to quantify the level of controversy for a given topic.
3.2 Background: Theoretical Models to Define Controversy

While there has been little work toward developing theoretical models in the domain of controversy, we introduce two related efforts that have modeled controversy. Dori-Hacohen (2017) presented a theoretical model to define controversy within a group of people, or a population. Her model is inspired by growing disparity between scientific understanding and public opinion on certain controversial topics, such as climate change, evolution, and vaccination. While many scientists think that there is no controversy with regard to those topics, in a general population, non-scientific claims and arguments proliferate causing the topics to be highly controversial. Hence, she argues that controversy is not a global and static value for a topic, but rather defined by a function that takes a given population as well as the topic.

Let $\Omega$ be a population of $n$ people. Let $T$ be a topic of interest to at least one person in $\Omega$. Her model assumes that controversy is a multi-dimensional factor of traits that can be observed in $\Omega$. She hypothesizes that such dimensions include contention to measure how contentious the topic is, importance to measure how important the topic is to people, and conviction to encode who strongly holds their belief in their stances as follows:

$$\text{controversy}(\Omega, T) = f(\text{contention}(\Omega, T), \text{conviction}(\Omega, T), \text{importance}(\Omega, T))$$

Dori-Hacohen defines the probability of contention within a population as the probability of randomly drawing two people that have different stances that are in conflict with each other on a given topic. While she modeled “contention” in her work, she left other dimensions unexplored. In work outside of this proposal, we explored the dimension of “importance” by suggesting that the importance of the topic should also be measured with regard to the population, specifically by the ratio of people who are affected by $T$ in $\Omega$ (Jang et al., 2017). This was measured by counting people who post tweets on the topic at least once during the time of observation.
Zielinski et al. (2018) also recognized the necessity of having a theoretic model that formally defines controversy. Their formal is based on a Merriam-Webster dictionary definition of controversy as “argument that involves many people who strongly disagree about something: strong disagreement about something among a large group of people.” Their proposed function takes three variables, a given object \( d \) (e.g., a document), a given community \( \Omega \), and an empirical distribution of opinions given by members in \( \Omega \) in \( d (E_{\Omega}^d) \), to output a binary classification as follows:

\[
f(d, \Omega, E_{\Omega}^d) = \{ \text{controversial, non-controversial} \}
\]  

(3.1)

Although they used a slightly different terminology such as referring to population in Dori-Hacohen’s model as community, the underlying assumption of the model captures the same intuition that “contention” within a given set of people is the main feature to measure controversy of a given object or topic itself. In this proposal, we will use the term “population”.

### 3.3 A Probabilistic Model of knn-WC Algorithm

In this section, we analyze and derive a model for the kNN-WC algorithm. We stress that our goal is not to design a new model, but to propose a theoretical model that explains the kNN-WC algorithm. We first summarize the framework as the four steps to make a controversy classification on general webpages:

1. **Matching k-NN Wikipages:** When a webpage is given as an input, they find \( k \) nearest-neighbor Wikipages by generating a query of keywords extracted from the document.

2. **Computing controversy score on Wikipages:** From each of the \( k \) Wikipages, they automatically computed three controversy scores: \( C \) score (Das et al.,
2013b), M score (Yasseri et al., 2012), P score (Dori-Hacohen, 2017). In addition to these, they extracted D score that is a binary score that indicates the presence of Dispute tags assigned by Wikipedia editors (Kittur et al., 2007).

3. **Aggregate:** They aggregated the multiple scores of $k$ Wikipages using average or max operators.

4. **Vote and classify:** They apply a voting scheme to turn the aggregated scores into a final binary decision, controversial or non-controversial.

Let us define a probabilistic framework that explains those steps by estimating the probabilistic components. Let $D$ be the text of document, and $T$ be the topic of the document $D$. In this model, a topic is defined a Wikipedia page (Wikipage) including its meta-data such as edit history. For example, $T$ would be the most relevant Wikipage to $D$ from the set $W$ that contains all possible topics (i.e., Wikipages). We will interchangeably use the term topics and Wikipages from this point.

Finally, we define $C$ be the binary variable to denote the controversiality of $D$. $P(C = 1|D)$ indicates that $D$ is controversial, and $P(C = 0|D)$ means the opposite. For simplicity, we define the constant variable $c$ to denote $C = 1$ and represent the query probability in a concise form: $P(c|D)$ to denote the probability that $D$ contains controversiality and $P(nc|D)$ to mean that $D$ does not contain controversiality (i.e., contains non-controversiality). The model aims to estimate $P(c|D)$ to determine whether or not the given document $D$ contains controversiality. We summarize the notations used in our modeling in Table 3.1.
Table 3.1: A summary of notations used in our probabilistic framework

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>A document text consisting of words</td>
</tr>
<tr>
<td>$T$</td>
<td>A topic of $D$. In this model, a Wikipage.</td>
</tr>
<tr>
<td>$W$</td>
<td>A set of all topics. In this model, Wikipedia pages</td>
</tr>
<tr>
<td>$C$</td>
<td>A binary variable to denote a $D$ contains a controversiality</td>
</tr>
<tr>
<td>$c$</td>
<td>A constant to denote that $C = 1$</td>
</tr>
<tr>
<td>$P(c</td>
<td>D)$</td>
</tr>
<tr>
<td>$P(nc</td>
<td>D)$</td>
</tr>
<tr>
<td>$\Omega_w$</td>
<td>A set of Wikipedia editors who contributed Wikipage $w$</td>
</tr>
<tr>
<td>$q_D$</td>
<td>A query generated from $D$</td>
</tr>
</tbody>
</table>

First, we interpret “$D$ containing controversiality” to mean that $D$ discusses a controversial topic. $P(c|D)$ can be obtained from a marginal probability of the joint probability $P(c, D, T)$ for all possible topics of $T$ in $W$.

$$P(c|D) = \frac{P(c, D)}{P(D)} = \frac{\sum_{w \in \Omega} P(c, D, T = w)}{P(D)} \quad (3.2)$$

Because the probabilities of $P(c|D), P(T|D)$, and $P(c|T)$ are closely associated with each other, we represent their relationship with a probabilistic graphical model that has three random variables, $D, T,$ and $C$. We capture the following algorithm’s flow by constructing a linearly-structured Bayesian network as shown in Figure 3.1: the topics ($T$) are determined by the query from the document ($D$), the controversiality ($C$) is determined by the contention level of topics. Intuitively, if the topic of the document is known, controversiality can be derived from that topic, which explains why $C$ and $D$ are conditionally independent given $T$. Based on the network, a joint probability distribution, $P(c, D, T)$ is defined as follows:

$$P(c, D, T) = P(c|T) \cdot P(T|D) \cdot P(D) \quad (3.3)$$

Finally, we derive $P(c|D)$ from Eq. (3.2) and Eq. (3.3). $P(c|D)$ is broken down to two components, the probability that a given document $D$ retrieves a topic $T$, and
the probability that $T$ is controversial. For estimating $P(w|D)$, instead of considering all of $D$, they generate a query $q_D$ from $D$ to retrieve $w$. In addition, instead of considering all Wikipedia pages to aggregate the probabilities from, they take the top $k$ most relevant Wikipages and estimate the probabilities from them. Let the top $k$ most relevant Wikipages of $D$ as $W_D$:

$$P(c|D) = \sum_{w \in W_D} \left[ P(c|w) \cdot P(w|q_D) \right]$$  \hspace{1cm} (3.4)

### 3.3.1 Estimating $P(c|w)$ using Contention

In our model, $P(c|w)$ indicates the probability that a given Wikipage $w$ is controversial. There has been some work that focused on estimating the level of controversy of Wikipages. For the $k$NN-WC algorithm, three state-of-the-art techniques (Yasseri et al., 2012; Dori-Hacohen, 2017; Das et al., 2013b) as well as binary dispute tags that are manually-curated by Wikipedia editors have been tested. We define them as Wikipedia Controversy Features (WCF). Among these, M score has been shown to be most effective for their framework. Therefore, we discuss M score as well as P score that captures the same intuition as M score but that is derived from a more probabilistic grounds in depth.

P score (Dori-Hacohen, 2017) and M score (Yasseri et al., 2012) both measure controversy as the level of “contention” within a group of people. This viewpoint is proposed to define controversy by the population model. While P score is an application of the population model in Wikipedia, the viewpoint retrospectively explains the intuition behind M score.

Recall that the population model argues that the level of controversy of a given topic can only be answered with respect to a given population, and specifically, with regards to how contentious the topic is within the population. Both P score and

---

1The controversy-population model was proposed 5 years later than the M score.
M score assume that a population was given as the set of Wikipedia editors who contributed to the given topic. We explicitly transform the query $P(C|T)$ to an equivalent population-aware query by treating $\Omega_w$, a population of Wikipedia editors on Wikipage $w$ as a given parameter when $w$ corresponds to the topic $T$.

$$P(C|T) = P(C|w) = P(Contention|w; \Omega_w) \quad \text{(3.5)}$$

To estimate the level of contention, M score and P score both use “mutual reverts”, online activities of Wikipedia editors where two editors have reverted each other’s contribution, as a sign of disputes. The common intuition that both measures try to capture is that the contention increases as there are more reliable mutual reverts.

We first denote a set of Wikipedia editors that have contributed to a Wikipage $w$ as $\Omega_w = \{p_1, p_2, ... p_n\}$. We define $\text{mutualrevert}(p_i, p_j)$ as a binary relationship that indicates whether reviewers $p_i$ and $p_j$ have mutually reverted each other. However, not all mutual reverts are meaningful. Vandalism is an act of maliciously editing Wikipages. Some mutual reverts are caused to fix these malicious activities, and should not be counted towards measuring contention.

Let $MR_D = \{(p_i, p_j) | p_i, p_j \in \Omega_w, \text{s.t.}, i < j \land \text{mutualreverts}(p_i, p_j)\}$ be the set of unique pairs of editors that have mutually reverted each other on $D$. Sumi et al. (2011) define $N_{p,D}$ be a reputation score of editor $p$, which indicates how credible $p$ is, without going into much details here. The higher the reputation score is, the less likely that $p$ to be a vandal.

**M Score:** To estimate if a given mutual revert is not caused by vandalism, they use a heuristic, $\text{min}(N_{p_i,D}, N_{p_j,D})$, to indicate how unlikely it is that any of the editors are vandals. M Score is computed as follows:

$$M = |\Omega_w^R| \cdot \sum_{(p_i, p_j) \in MR_D} \text{min}(N_{p_i,D}, N_{p_j,D}) \quad \text{(3.6)}$$
where $\Omega_{w}^R$ is a sub-population of $\Omega_{w}$ that is involved in at least one mutual revert that occurred in $w$.

**P Score:** Dori-Hacohen (2017) defines P score as the probability of drawing two random editors and the two editors have a mutual revert. Each mutual revert is discounted by the probability that each editor is not a vandal:

$$P = \frac{1}{|\Omega_w|^2} \cdot \sum_{(p_i, p_j) \in MR_D} \frac{N_{p_i,D}}{N_{max}^{D} + 1} \cdot \frac{N_{p_j,D}}{N_{max}^{D} + 1}$$

(3.7)

where $N_{max}^{D}$ is the maximum reputation score of any editor who contributed to $D$.

While P score can be directly used as a probability of contention, M score is not a probability, but an unbounded integer. We convert M score as a probabilistic score namely $P_M$ by normalizing by the maximum M score among all Wikipedia pages. By using the estimated probability from P or M score, we can develop the model as follows:

$$P(c|D) = \sum_{w \in W_D} [P_M(contention | w; \Omega_w) \cdot P(w|q_D)]$$

(3.8)

where $P_M$ is a normalized M score.
3.4 Discussion

From deriving the model, we suggest that the existing algorithm rely on following two assumptions, and the empirical success of the algorithm depends on how realistic these assumptions are:

- **A.1:** A query generated from the document retrieves Wikipages that represent the document’s topics.

- **A.2:** Wikipages that discuss controversial topics will show a high level of contention among the editors of the page.

In addition, the model also reveals following two properties.

- **P.1:** The model does not directly model non-controversiality. For example, having more non-controversial content in the document does not necessarily decrease the level of controversy.

- **P.2:** Finding the evidence of “dispute” between people is a necessary condition for identifying controversy on a given topic.

In the next chapter, we will revisit the algorithm to investigate how reasonable these assumptions are. We also challenge the identified properties to explore a new model.
As discussed in the previous chapter, the $k$NN-WC algorithm can be viewed as an instantiation of the probabilistic model presented in Eq. 3.8. In this chapter, we first revisit the $k$NN-WC algorithm and analyze the weaknesses of the algorithm by challenging two assumptions identified that the algorithm is built on. We then propose solutions to fix them to improve the performance of the algorithm.

### 4.1 Revisiting the assumptions of the algorithm

We argue that the $k$NN-WC often fails because the two assumptions (A.1 and A.2) are not satisfied in their empirical performance. We explain why these assumptions are often wrong in the algorithm and how to address them to fix.

#### 4.1.1 The Limitation of a Single Document Query

**A.1: A query generated from the document retrieves Wikipages that represent the document’s topic.**

To identify $k$ relevant Wikipages, $k$NN-WC generates a query from each document for Wikipedia retrieval. To generate a query for the document (i.e., a document query), $k$NN-WC takes a straightforward solution for generating a document query is simply to use the “best” $k$ keywords. However, generating the global keyword query from the document has two issues. First, as the document almost always contains multiple sub-topics, the generated query contains an unknown mixture of different sub-topics. This makes the query’s intent less clear, as it targets many sub-topics.
at the same time and in unknown balance. Second, it is unlikely that all sub-topics are covered in the query – or covered appropriately – because keywords are extracted from a bag-of-words, which does not model the existence of sub-topics as it is. We address these weaknesses by generating multiple queries from the document that cover different topics, and aggregating multiple ranked lists from each query. We discuss this approach in the next section.

4.1.2 The Limitation of Wikipedia Controversy Features

A.2: Wikipages that discuss controversial topics will show a high level of contention among the editors of the page.

To estimate $P(\text{Contention}|w; \Omega_w)$, Dori-Hacohen and Allan examined the previous work that study the signals of “controversy” in Wikipedia (Kittur et al., 2007; Das et al., 2013a; Yasseri et al., 2012). We refer these scores as Wikipedia Controversy Features (WCF). The algorithms used various information extracted from Wikipedia pages, meta-data, talk pages, and the page’s edit history. Note that the authors of the previous work used the term “controversy” in their work, because they mainly focus on the dispute signals between Wikipedia editors, we interpret that “contention” and “controversy” refer to the same concept in this context.

$k\text{NN-WC}$ uses the WCF to estimate $P(\text{Contention}|w, \Omega_w)$ because WCF models “edit-war” features, evidence of multiple editors ($\Omega_w$) exchanging opposing opinions on the given Wikipage ($w$). We introduce the three features that Dori-Hacohen and Allan used in their algorithm, which we also use for realization of our new model later:

C score This score was generated by a regression-based method (Kittur et al., 2007) that estimates the revision count of controversial Wikipedia pages, which are labeled with \{controversial\} tags. The algorithm was trained with the edit-history information, such as number of unique editors and number of reverts,
as well as some metadata of Wikipedia pages. The score is normalized so that it ranges between 0-1.

**M score** Another controversy score generated by studied by Yasseri et al. is based on statistical features of edits, which signify how fierce the “edit war” is (Yasseri et al., 2012). The statistical features include the number of mutual reverts of two editors, the number of editors participating in this edit-war, and the editor’s reputation. M score is theoretically unbounded ranging from 0 to a few billions.

**D score** This is a Boolean value indicating whether a Wikipedia page contains a *dispute* tag in it. This tag is assigned by the page’s contributors if the Wikipage’s talk page shows some level of disputes. Unlike the above two scores, this label is manually curated. Hence, this score is extremely sparse; only 0.03% of the articles have a positive D score (Kittur et al., 2007).

Unfortunately, these approaches are limited for the same reason that many Wikipages with controversial topics do not have sufficient edit-history or explicit edit-wars. There is a tendency that the heat of the edit-wars are focused on one Wikipage of a general and broad topic, leaving other related but sub-topical pages less attended. After all, there is simply no point of having the same “war” on all similar Wikipages. Table 4.1 shows an example of a few “abortion” related topics and their M and C score. While the “Abortion” page received a lot of attention, other pages with more specific topics such as *Abortion in certain countries* and *Abortion Act* had virtually no edit-wars. Unless there is a specific issue or event specifically tied to the page, all general disputes on abortion have been delegated to the “Abortion” page. In other words, not having the “edit-war” does not necessarily mean that there was no war in this topic, but that the war has been happening somewhere else instead. This phenomenon causes the algorithm to easily make false negative errors (i.e., classifying “controversial” as “non-controversial”) as illustrated in Table 4.1).
Table 4.1: An example of M score and C score for Wikipages on “Abortion” that most sub-pages on “Abortion” have controversy scores close to 0.

<table>
<thead>
<tr>
<th></th>
<th>M score</th>
<th>C score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abortion</td>
<td>4,102,593</td>
<td>0.300</td>
</tr>
<tr>
<td>Abortion_Act</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Abortion_in_China</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Abortion_in_England</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Abortion_in_the_US</td>
<td>0</td>
<td>0.002</td>
</tr>
</tbody>
</table>

4.2 Proposed Framework

To address the two drawbacks of the algorithm, we propose two modifications to the existing framework.

4.2.1 TileQuery Generation

4.2.1.1 Document Segmentation

We first use the block comparison algorithm described by the TextTiling technique (Hearst, 1997). The block comparison method defines as block with a few sentences, and computes a lexical similarity score for every gap between two blocks. When the similarity score dramatically changes at a gap, we assume that is where a sub-topic shift occurs and create a tile of blocks.

4.2.1.2 Query Generation

Once we create tiles from a document using TextTiling, we generate a query from each tile. There are often some tiles that are hard to understand its meaning without the context of the full text. Therefore, adding the global context helps clarify the topic of each tile, anchoring the tile’s query to the containing document’s topic. We hence generate a query by using the $g$ (global) most frequent terms from the document, and the $l$ (local) most frequent terms from the tile. We empirically found that $g = 3, l = 7$ gives the best performance when using 10 terms.
4.2.1.3 Aggregating the Ranked Lists

Each tilequery returns a ranked list. We compute the relevance score for each retrieved Wikipage $w_i$ by aggregating the reversed ranking order:

\[
Relevance(w_i) = \sum_{t_i \in T} (k - rank_{t_i}(w_i))
\] (4.1)

where $T$ is the set of tiles generated from the document. This scoring prioritizes Wikipages that appear high in some tile or at reasonable ranks in many tiles, or preferably both.

4.2.2 Smoothing Controversy score of Wikipages

Previous work studied algorithms for automatically computing scores that estimate the level of controversy. They use features available in Wikipages, meta-data, talk pages, and edit-history. We briefly explain the three scores that the previous framework adopted.

Due to this phenomenon, even if we generate a better query to find more relevant $k$ pages, the framework still would not be able to fully take advantage of that due to the underestimated scores. Hence, it is necessary to revise these scores to reflect controversy better. If the purpose of the M or C score was to measure the controversy level presented in the Wikipage per se, we need newly revised scores that accurately signify controversiality of the topic in general. To do this, we will construct a network that links topically related neighbors within the Wikipedia. We then revise the controversy score by “smoothing” using the scores of neighbors with more edit history, whose scores were computed with more confidence.
4.2.2.1 Constructing Wikipages’ Graph

We construct a tree-structured graph to identify topically related neighbors of a Wikipage. Let $G = (V, E)$ be a directed graph with nodes $V$ (Wikipages) and edges $E$ (sub-topical relations). An edge $e(u, v)$ represents that node $v$ is a sub-topic of $u$.

As a simple and straightforward method to construct the edges, we look at their titles. If a Wikipage $u$’s title is used as a prefix of other $v$’s title, we assume that $v$ is sub-topic of $u$. While we use nodes’ titles to construct edges, we assume there is a mapping between a title and a node and will use them interchangeably.

![Figure 4.1: An example of the constructed graph for Abortion and two different sub-graphs selected based on the two methods.](image)

Let a Wikititle $T$ be a ordered list $[t_1, t_2, ..., t_n]$, where $t_i$ is an i-th space-delimited token. The parent node set $P(T)$ (i.e., Wikipages whose titles that have $T$ as a child) is obtained by:

$$P(T) = \{P_T^i | P_T^i \in W_T, i \in \{1..n\}\}, P_T^i = concatenate[t_1, ..., t_i]$$

where $W_T$ is a set of all Wikipedia titles. The graph also contains many noisy relations when the prefix is an ambiguous entity, or a simply too general word, such as “American”. To filter out the noisy relations, we remove the edges if two pages are not linked in any direction. Using this graph, we finally revise the controversy score.
using smoothing. Figure 4.1 shows an example of constructed graph for the topic of “Abortion”.

4.2.2.2 Graph-based Smoothing

When Wikipage is given as a query, we extract a sub-graph around the node from the constructed graph using one of the two methods, whose examples are demonstrated in Figure 4.1:

- **Pair-based**: A sub-graph around the query node including its children at all depth and its parents. The resultant graph only consists of nodes that have a direct prefix-contain relation with the query node.

- **Clique-based**: In addition to the sub-graph obtained by the pair-based method, sibling nodes that share the same parents with the query node are added. Although siblings may not be topically related to the query node especially if the parent (i.e., prefix) is a general term, this allows broader coverage of potentially related pages.

Once we obtain the sub-graph, we treat all nodes in the sub-graph as topically related neighbors of the query node. We want to fix the query node’s controversy score by smoothing from neighbors that have more reliable scores. For that we assume that the controversy score of a Wikipage with more revision history is more reliable. Hence we convert this graph into a weighted, directed network whose direction represents which way influence should extend to (i.e., the one with higher revision count to the other with lower count), and whose edge weight represents the confidence of the influence relation, which is the revision count of the source (Figure 4.2). From the graph, the new controversy score of Wikipage $W_i$ is computed as:

$$C'(W_i) = \sum_{W_j \in \text{links}(W_i)} \frac{C(W_j) \times r_j}{\sum r_k} \quad (4.2)$$
where \( r_i \) is a revision count of \( W_i \).

4.2.3 Aggregation and Voting

We summarize the aggregation and voting schemes introduced by previous work. Once the controversy scores are obtained for \( k \) Wikipages, we aggregate the \( k \) scores by taking average or max of them. Since we use three different scores, M, C, and D, three aggregated scores, \( M_{agg} \), \( C_{agg} \), and \( D_{agg} \) are computed. We turn these scores into binary label indicating controversial (1) or non-controversial (0), using corresponding thresholds. \( M_{label} = 1 \) if \( M_{agg} \geq \text{Threshold}_M \), and 0 otherwise. Using the three generated labels, we use a voting scheme to make a final decision. We test 6 voting schemes as parameters in our experiments.

The webpage is controversial if:

- **C/M/D**: \( \{C_{label}, M_{label}, D_{label} \} \) is 1, respectively.

- **Majority**: the majority (i.e., at least two) of \( \{C_{label}, M_{label}, D_{label} \} \) is 1.

- **Or/And**: \( C_{label} \{\lor/\land\} M_{label} \{\lor/\land\} D_{label} \) is 1.
Table 4.2: An example of two controversy scores on several Wikipages on “Abortion”, before and after score smoothing

<table>
<thead>
<tr>
<th></th>
<th>Original scores</th>
<th>Revised scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M C</td>
<td>M C</td>
</tr>
<tr>
<td>Abortion</td>
<td>4,102,593 0.300</td>
<td>4,102,593 0.300</td>
</tr>
<tr>
<td>Abortion_Act</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Abortion_in_China</td>
<td>0 0</td>
<td>2,062,156 0.166</td>
</tr>
<tr>
<td>Abortion_in_England</td>
<td>0 0</td>
<td>2,128,909 0.172</td>
</tr>
<tr>
<td>Abortion_in_the_US</td>
<td>0 0.002</td>
<td>2,983,300 0.218</td>
</tr>
</tbody>
</table>

Table 4.3: Accuracy, F1, and the best parameters in 5-fold runs for different query and inferred score settings.

<table>
<thead>
<tr>
<th>Run</th>
<th>Query</th>
<th>Inferred score</th>
<th>K</th>
<th>C Threshold</th>
<th>M Threshold</th>
<th>Aggregation</th>
<th>Acc.</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/A</td>
<td>5, 20</td>
<td></td>
<td>0.17, 4.18 · 10⁻²</td>
<td>10000, 20000</td>
<td>Maj.</td>
<td>0.72</td>
<td>0.50</td>
</tr>
<tr>
<td>2</td>
<td>ALL</td>
<td>Clique 15</td>
<td></td>
<td>0.17, 4.18 · 10⁻²</td>
<td>10000, 20000</td>
<td>Maj.</td>
<td>0.78</td>
<td>0.68</td>
</tr>
<tr>
<td>3</td>
<td>Pair</td>
<td>5, 20</td>
<td></td>
<td>0.17, 4.18 · 10⁻²</td>
<td>10000, 20000</td>
<td>Maj.</td>
<td>0.73</td>
<td>0.53</td>
</tr>
<tr>
<td>4</td>
<td>N/A</td>
<td>20</td>
<td></td>
<td>4.18 · 10⁻²</td>
<td>20000, 40000, 84930</td>
<td>Maj.</td>
<td>0.75</td>
<td>0.57</td>
</tr>
<tr>
<td>5</td>
<td>TF10</td>
<td>Clique 20</td>
<td></td>
<td>4.18 · 10⁻²</td>
<td>84930</td>
<td>Maj.</td>
<td>0.79</td>
<td>0.68</td>
</tr>
<tr>
<td>6</td>
<td>Pair</td>
<td>10, 20</td>
<td></td>
<td>4.18 · 10⁻²</td>
<td>20000, 84930</td>
<td>Maj.</td>
<td>0.75</td>
<td>0.57</td>
</tr>
<tr>
<td>7</td>
<td>N/A</td>
<td>10,15,20</td>
<td></td>
<td>4.18 · 10⁻²</td>
<td>10000, 20000</td>
<td>Maj.</td>
<td>0.75</td>
<td>0.59</td>
</tr>
<tr>
<td>8</td>
<td>TILE</td>
<td>Clique 20</td>
<td></td>
<td>0.17</td>
<td>40000</td>
<td>M</td>
<td>0.80</td>
<td>0.71</td>
</tr>
<tr>
<td>9</td>
<td>Pair</td>
<td>10,15,20</td>
<td></td>
<td>4.18 · 10⁻²</td>
<td>10000, 20000</td>
<td>Maj.</td>
<td>0.75</td>
<td>0.61</td>
</tr>
</tbody>
</table>

4.3 Experiments

4.3.1 Dataset

We use the publicly available controversy dataset\(^1\) released by Dori-Hacohen and Allan (2013). The dataset consists for 303 webpages from the ClueWeb09 collection, which is a publicly available dataset of crawled general webpages\(^2\). Note that the annotated webpages do not include any Wikipages. Each document is annotated with the controversy level of four scales: 1 - “clearly controversial”, 2 - “possibly controversial”, 3 - “possibly non-controversial”, and 4 - “clearly non-controversial”.

To convert the annotations to binary judgments, we treated the documents with average ratings among annotators of less than 2.5 as controversial, and otherwise non-controversial. Of 303 documents, 42% of them are controversial.

\(^1\)http://ciir.cs.umass.edu/downloads
\(^2\)http://lemurproject.org/clueweb09/
Table 4.4: Improvements of accuracy and F1 score between runs (bold: statistically significant)

| Row # | Runs       | |Acc\_1\-Acc\_2| |F1\_1\-F1\_2| | p value |
|-------|------------|----------------|----------------|----------------|----------------|
| 1     | 1 vs 2     | 6%             | 2%             | 0.01 \cdot 10^{-2} |
| 2     | 1 vs 4     | 3%             | 7%             | 0.61           |
| 3     | 1 vs 7     | 3%             | 9%             | 0.08           |
| 4     | 4 vs 5     | 4%             | 11%            | 0.17 \cdot 10^{-2} |
| 5     | 4 vs 7     | 0%             | 2%             | 0.18           |
| 6     | 5 vs 8     | 1%             | 3%             | 0.06 \cdot 10^{-2} |
| 7     | 6 vs 9     | 0%             | 4%             | 0.01 \cdot 10^{-2} |
| 8     | 7 vs 8     | 5%             | 12%            | 0.01 \cdot 10^{-2} |

To test the effectiveness of the proposed query method, we consider two other query baselines. One is TF10, the 10 most frequent terms, as in the prior work. As taking only $k$ terms as in a query might miss information, we consider another baseline, all query that uses all terms in a document as a query to observe the extreme case of TFN.

We consider 9 settings from all possible combinations of three query methods and three scoring schemes (Table 4.3). Run 4 is the setting proposed in the prior work (Dori-Hacohen and Allan, 2015). In each setting, we varied the four sets of parameters, the number of neighbors $K$ (1, 5, 10, 15, 20), aggregation method (avg, max), voting methods (C, M, D, Majority, Or, And, D ∨ (C ∧ M)), and thresholds for C and M as tested in the prior work. We found the best parameter setting for each run using 5-fold cross validation with the target metric accuracy. Thus, for 9 settings, there are 5 sets of parameters learned for each fold. We used McNemar’s Test$^{3}$ for statistical significance test.
4.3.2 Results and Discussion

Our statistical significant tests suggest that the difference of accuracies between the three query methods in runs 1, 4, and 7 are not significant (Row 2 & 5 in Table 4.4), which suggest that the three methods mostly made similar classifications. However, once we apply neighbor-based smoothing on controversy scores, query methods cause classification to work differently. The accuracy gain of TileQuery over TF10 was 1%, and 4% of F1-score with smoothing. Although the accuracy gain was small, the query set that each method performed well was different as the significance test implies (Row 6 & 7 in Table 4.4).

In all settings, using controversy score smoothing significantly improved the classification accuracy and F1-score. As row 1, 4, and 8 in Table 4.4 show, the accuracy was improved by 4-6% and the F1-score was improved by 2-12% in all three query methods. Clique-based neighbor selection consistently outperformed pair-based selection.

4.4 Conclusion

We revisited the prior work for automatically detecting controversy from the general open-domain webpages. We identified two major weakness in the framework and proposed two modifications to fix the issues. The controversy score smoothing consistently improved the controversy classification accuracies by 4-6% compared to those without smoothing. Overall, the run with our two modifications of TILEQUERY and controversy score smoothing gave the best accuracy improving the previous framework by 5%.

\[^3\text{https://en.wikipedia.org/wiki/McNemar\%27s\_test}\]
CHAPTER 5
CONTROVERSY LANGUAGE MODELS

5.1 Introduction

We showed how the state-of-the art $k$NN-WC algorithm is based on the probabilistic model and identified two properties of the model in Chapter 3. In this chapter, we start by revisiting and challenging the properties to propose a new model for controversy detection that has complementary properties.

Property 1: Non-controversiality of a document is not directly modeled.

The $k$NN-WC model does not directly consider the probability that a document is non-controversial. This means that when a document contains more non-controversial keywords, it does not directly decrease the probability of controversy because the probability of controversy is more affected by the presence of controversial keywords. Instead of defining non-controversiality simply as a lack of controversy signals, could modeling non-controversiality separately make the controversy classification more robust?

Property 2: Evidence of contention within a population is the only signal to identify controversy.

We asserted in Chapter 4 that the “contention” signal is not reliable because it is selectively available. The disputes are expensive signals in the sense that they require multiple people to engage in the discussion. Not only they are sparsely available, but also their presence can easily be delayed until enough people generate a contentious discussion. This model is inherently limited in efficacy and adaptability because “contention” exists sparingly and takes some time to be observed.
Therefore, we explore another probabilistic model of controversy to challenge the above properties shown in the previous model. We aim to use an alternative “language” signal and also directly model non-controversiality for the controversy classification. As part of our effort to find a new model for controversy detection, we first turn to social science research to understand how controversy is being identified and shaped. The most relevant work to our interests would be Cramer’s (2011). Cramer explains that “controversy” cannot necessarily be verified to exist in the world independent of its appearance in text, but rather it is created and shaped by the discourse surrounding it, particularly in news outlets. He refrains from defining the term directly, referring to it as a “metadiscursive” (terms that are used to denote a discussion of discussion) and “indexical” (terms whose specific meaning changes from context to context) term, meaning that it may be difficult to formulate a mathematical or technical definition of controversy, and it can be loosely defined as something that you would know when you see it. Cramer’s work suggests that language could be a key feature in identifying controversy.

5.2 Proposed Model

Cramer manually studies patterns of text surrounding specific terms such as controversy, dispute, scandal, and saga within the Reuters corpus (Rose and Whitehead, 2002), as being indicative of controversy. Motivated by Cramer’s research, we propose a new probabilistic model of controversy that considers how similar the document’s language is to the one that discusses a range of controversial topics.

Recall that $P(c|D)$ indicate the probability that $D$ is controversial, and $P(c|D) + P(nc|D) = 1$. For the purpose of binary classification, we are only interested in whether $P(c|D) > P(nc|D)$ holds, rather than the actual probabilities, so we can use rank-safe approximations.
Each of $P(c|D)$ and $P(nc|D)$ can be represented using Bayes’ theorem, which allows us to consider the following odds-ratio:

$$\frac{P(c|D)}{P(nc|D)} = \frac{P(D|c)}{P(D|nc)} \cdot \frac{P(c)}{P(nc)} > 1$$  \hspace{1cm} (5.1)

Now our test condition can be expressed as:

$$\frac{P(D|c)}{P(D|nc)} > \frac{P(nc)}{P(c)}$$ \hspace{1cm} (5.2)

where for our purposes, we can treat the right hand side as a constant threshold (since it is independent of the document $D$), which can be learned with training data. To avoid underflow, we actually calculate the log of this ratio.

$$\log P(D|c) - \log P(D|nc) > \alpha$$ \hspace{1cm} (5.3)

Therefore, we only have to estimate the probabilities $P(D|c)$ and $P(D|nc)$, which we do using the language modeling framework by the construction of a language model of controversy $L_C$, and a non-controversial language model $L_{NC}$. We make the standard term independence assumption for each word ($v$) in our document ($D$), and avoid zero probabilities with linear smoothing. We create another language model $L_G$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_C$</td>
<td>A language model of controversy</td>
</tr>
<tr>
<td>$L_{NC}$</td>
<td>A language model of non-controversy</td>
</tr>
<tr>
<td>$L_G$</td>
<td>A background language model of all topics</td>
</tr>
<tr>
<td>$D_C$</td>
<td>A set of controversial documents used to build $L_C$</td>
</tr>
<tr>
<td>$D_{NC}$</td>
<td>A set of controversial documents used to build $L_{NC}$</td>
</tr>
<tr>
<td>$tf(w, D)$</td>
<td>The frequency of term $w$ in a document $D$</td>
</tr>
<tr>
<td>$P(w</td>
<td>L)$</td>
</tr>
</tbody>
</table>
for the purpose of smoothing using a broad “background” collection of documents, as opposed to controversial and non-controversial collections. In practice, we estimate both the general language model ($L_G$) and the non-controversial language model ($L_{NC}$) as the same by constructing them from the set of all documents.

$$P(D|c) \approx P(D|L_C) = \prod_{v \in D} (\lambda P(v|L_C) + (1 - \lambda) P(v|L_G))$$

$$P(D|nc) \approx P(D|L_{NC}) \approx P(D|L_G) = \prod_{w \in D} P(v|L_G)$$

Here, $D_C$ is a set of controversial documents, and $D_{NC}$ is a set of non-controversial documents, which we estimate in our collections as the background collection, $D_{BG}$.

$$P(w|L_C) = \frac{\sum_{d \in D_C} tf(w,d)}{\sum_{d \in D_C} |d|}, P(w|L_{NC}) = \frac{\sum_{d \in D_{BG}} tf(w,d)}{\sum_{d \in D_{BG}} |d|}$$

Therefore, to build a language model of controversy, we need to find $D_C$. We explore Wikipedia Controversy Features (WCF) and Cramer-inspired query based models to construct $D_C$ as following:

- **Highly Contentious Articles** While the normalized WCF features are used to estimate $P(Contention|w; \Omega_w)$ in the probabilistic model of the kNN-WC, we simply take the top $K$ articles that have high WCF values in Wikipedia. In our experiments, three types of WCF, M/C/D scores are considered.

- **Controversy-indicative terms**: Documents that are retrieved by a query believed to indicate controversy. We explore Cramer’s terms as well as manual lexicons from past work (Mejova et al., 2014; Roitman et al., 2016). The examples of these terms is shown in Table 5.2.
Table 5.2: An example of controversy-indicative terms.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Search Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roitman et al.</td>
<td>dispute, disputable, disagreement, debate, polemic, feud, question, schism wrangle, controversy, dispeace, dissension, criticism, argue, disagree, claim argument, conflict, opposition, adversary, antagonism, oppose, object, case, loggerheads, quarrel, fuss, moot, hassle, altercate, evidence, clash, issue, problem, emphasize, recommend, suggest, assert, defend, maintain, reject, support, challenge, doubt, refute, confirm, prove, validate, establish, concur substantiate, verify, against, resist, support, agree, consent, accept, refuse plead, right, justify, justification</td>
</tr>
<tr>
<td>Mejova et al.</td>
<td>abuse, administration, afghanistan, aid, american, army, attack, authority, ban, banks, benefits, bill, border, budget, campaign, candidate, catholic china, church, concerns, congress, conservative, control, country, court, crime, crisis, cuts, debate, debt, defense, deficit, democrats, disease, dollar, drug, economy, education, egypt, election, enforcement, fighting, finance, fiscal, force, funding, gas, government, gun, health, immigration, ...</td>
</tr>
<tr>
<td>Cramer et al.</td>
<td>controversy, dispute, saga, scandal</td>
</tr>
</tbody>
</table>

5.3 Evaluation

We leverage the same controversy dataset introduced in Chapter 4 that consists of judgments for 303 webpages. We perform 5-fold cross-validation and report measures on the reconstructed test set.

We implement the \(k\)NN-WC model as the baseline, both the original algorithm and the improved version of it introduced in Chapter 3. In order to construct \(D_C\), we needed the text of Wikipedia itself. Unfortunately, obtaining the same version of dumps as those used in prior work (Das et al., 2013a; Dori-Hacohen and Allan, 2015; Yasseri et al., 2012) is nearly impossible. For ease of future reproducibility, we leverage the long abstracts from the 2015-04 release of DBPedia\(^1\).

Prior work reported accuracy; we note that 65% of the 303 documents were non-controversial, so that accuracy does not provide the best view of this dataset. In this work, we primarily present results using the Area Under the Curve (AUC) measure, as we can compare performance without tuning thresholds. While AP and MAP have

\(^1\)http://wiki.dbpedia.org/
Table 5.3: The accuracy of the models. kNN-WC+ refers to the improved version of the algorithm presented in Chapter 4.

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN-WC model</td>
<td>0.737</td>
</tr>
<tr>
<td>kNN-WC+ model</td>
<td>0.800*</td>
</tr>
<tr>
<td>CLM</td>
<td>0.779</td>
</tr>
</tbody>
</table>

* This accuracy is not directly comparable to other rows because it was a reported number from a previous experiment that used a different version Wikipedia dump. I will include the direct comparison in the final thesis.

The same advantage for not requiring a threshold, AP explicitly gives advantages to a method that correctly predicts a few top-ranked items, which makes it more suitable for Information Retrieval tasks rather than classification tasks like ours (Su et al., 2015). Since accuracy was used in prior work, we report it as well in Table 5.3: Compared to kNN-WC algorithm, we improve from 0.72 accuracy (as reported by Dori-Hacohen and Allan (2015) and 0.737 accuracy (as reproduced) to 0.779 ($p < 0.001$). While we also report the accuracy of the improved version of the kNN-WC algorithm proposed in Chapter 4, note that these numbers are not directly comparable as we used a more recent version of Wikipedia dump as mentioned. As a result, the accuracy of the state-of-the-art baseline kNN-WC is reported as 0.750 while we report it as 0.737. We will include the direct comparison against the method in Chapter 4 for the final thesis. For our statistical significance tests, we follow in the footsteps of the pROC (Robin, 2014), and obtain confidence intervals from bootstrap resamples of the predictions.

For each fold, we trained two parameters by grid search: $K$, the number of top documents to choose, and $\lambda$, the smoothing parameter. For example, to create our M-score-based language model, we ranked the documents in our Wikipedia collection by their M score, and derived a language model based on the concatenation of the top $K$ documents. These models are presented in Table 5.4.
Table 5.4: Wikipedia-Based Controversy Detection Approaches. All Controversy Language Model (CLM) approaches have significant improvements over their respective \(k\)NN-WC counterpart at the \(p < 0.05\) level.

<table>
<thead>
<tr>
<th>Method</th>
<th>WCF</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)NN-WC model M</td>
<td></td>
<td>0.733</td>
</tr>
<tr>
<td>(k)NN-WC model C</td>
<td></td>
<td>0.743</td>
</tr>
<tr>
<td>(k)NN-WC model D</td>
<td></td>
<td>0.500†</td>
</tr>
<tr>
<td>CLM M</td>
<td></td>
<td>0.801</td>
</tr>
<tr>
<td>CLM C</td>
<td></td>
<td>0.835</td>
</tr>
<tr>
<td>CLM D</td>
<td></td>
<td>0.795</td>
</tr>
</tbody>
</table>

† In the \(k\)NN-WC-D approach, no neighbors were found with dispute tags, so it is equivalent to the weak baseline performance of the NO classifier.

Table 5.5: Language Models built from documents relevant to Cramer’s controversial terms (Cramer, 2011). Collection size \(|C|\) in millions of documents and type shown for comparison of results. We found that our wiki dataset was significantly better than all others, which had no pairwise differences otherwise.

| Expansion Dataset       | Type  | \(|C|\) | AUC  |
|-------------------------|-------|--------|------|
| DBPedia Wiki            | Wiki  | 4.6M   | 0.853|
| ClueWeb09B (Spam60) Web | Web   | 33.8M  | 0.741|
| Reuters News            | News  | 0.8M   | 0.745|
| NYT-LDC News            | News  | 1.8M   | 0.710|
| Robust04 News           | News  | 0.5M   | 0.711|
| Signal-1M News          | News  | 1M     | 0.710|

For building Cramer language models, where the relevant document sets were not created by WCF, we used the Galago search engine to rank documents using a query-likelihood retrieval. We explore 6 different corpora as document sources (Table 5.5). The \(K\) highest-scoring documents were then used as our controversial document set: \(D_C\).

5.4 Results

In Table 5.4, we present results of our models built around WCF. All our language modeling approaches are significantly stronger than the \(k\)-NN derived approaches. We only report results of WCF features independently because methods of aggregating
Table 5.6: Language Models built from Cramer’s terms and existing lexicons on DBPedia. We find that “controversy” is the most indicative term, and that “saga” is no better than random. Combining terms led to no improvement over “controversy” alone. “TF10” indicates that the TF10 query is used to represent a document whereas “Full” indicates that the full text of the document is used as a query.

<table>
<thead>
<tr>
<th>Query to build $D_C$</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>controversy</td>
<td>0.856</td>
</tr>
<tr>
<td>Roitman (Roitman et al., 2016)</td>
<td>0.823</td>
</tr>
<tr>
<td>dispute</td>
<td>0.740</td>
</tr>
<tr>
<td>scandal</td>
<td>0.721</td>
</tr>
<tr>
<td>Mejova (Mejova et al., 2014)</td>
<td>0.698</td>
</tr>
<tr>
<td>saga</td>
<td>0.500</td>
</tr>
</tbody>
</table>

Table 5.7: A comparison of lexicons built manually and through crowd-sourcing in prior work to our automatically derived language models A (*) indicates significant improvement over the best lexicon approach.

<table>
<thead>
<tr>
<th>Method</th>
<th>Document Query</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roitman Lexicon (Roitman et al., 2016)</td>
<td>TF10</td>
<td>0.543</td>
</tr>
<tr>
<td>Mejova Lexicon (Mejova et al., 2014)</td>
<td>TF10</td>
<td>0.562</td>
</tr>
<tr>
<td>Mejova Lexicon (Mejova et al., 2014)</td>
<td>Full</td>
<td>0.615</td>
</tr>
<tr>
<td>Roitman Lexicon (Roitman et al., 2016)</td>
<td>Full</td>
<td>0.695</td>
</tr>
<tr>
<td>Cramer Language Model</td>
<td>Full</td>
<td>0.783</td>
</tr>
<tr>
<td>WCF Language Model</td>
<td>Full</td>
<td>0.823*</td>
</tr>
<tr>
<td>WCF Language Model</td>
<td>TF10</td>
<td>0.835*</td>
</tr>
<tr>
<td>Cramer Language Model</td>
<td>TF10</td>
<td>0.856*</td>
</tr>
</tbody>
</table>
these features did not improve significantly over the best feature, and these methods were not quite comparable across kNN-WC and LM approaches.

In Table 5.5, we present an initial exploration of Cramer’s hypothesis that news is able to name and define controversy. While we were pleasantly surprised by the efficacy of this simple approach, we did not see the best performance in the news corpora (Rose and Whitehead, 2002) used by Cramer, but rather in using DBPedia as the expansion set. We also explored this approach on other news datasets (Robust04, NYT-LDC (Sandhaus, 2008), and Signal1M (Corney et al., 2016)) but results were statistically equivalent on all news corpora we tried.

While Cramer defined four keywords to be indicative of controversy, we find that “controversy” dominates effectiveness on this dataset. We explore these keywords as queries into an expansion corpus, and construct a language model from the highest scoring documents for the given query. That language model is then used for classification. Mejova et al. (2014) and Roitman et al. (2016) presented manually-curated lexicons for controversy tasks. We explore their use intrinsically, with Jaccard Similarity between the lexicon and the document terms in Table 5.7 and as queries to build a language model in Table 5.6.

5.5 Proposed Work: Investigating Generalizability

While the CLM approach showed a significantly better performance on controversy detection, its generalizability can be further investigated. Due to the limited scope of the dataset, we suspect that the maximum prediction accuracy that human can do on the same dataset. In other words, since even humans cannot do a better job than the current performance given their agreement ratio on the labels, achieving a higher accuracy on this dataset would be a meaningless goal (Foley, 2018).
To better understand the limitations of this approach, we further examine the model in terms of two aspects, generalizability and explainability. The specific research questions that we propose to investigate are as follows:

- **(Generalizability)** How generalizable is the CLM approach in predicting controversy for the documents that discuss recent controversial topics that were not included in the training corpus?

We plan to conduct an experiment of controversy detection with the newly published controversy dataset on Guardian News Articles published by Beelen et al. (2017) by training CLM with Wikipedia corpus that do not include the recent controversial topics. However, we might create a new dataset from Twitter that usually contains new online controversies, if necessary.

### 5.6 Proposed Work: Investigating Explainability of the Model

To understand the CLM approach beyond its classification performance, we also propose to analyze the explainability of the approach. We plan to analyze the model’s features by examining the keywords to the classification. Specifically, we will characterize the terms by a few categories, such as “indexical”, “topical”, and “sentimental” and analyze the characteristics of the feature terms that highly signal controversy (or non-controversiality). We also propose to qualitatively analyze how informative the top sentences selected by CLM can be to explain why the document is controversial or non-controversial.

- **(Explainability)** What are the characteristics of the keywords that lead the document to be controversial? Are the top keywords of a controversial document helpful keywords for users to understand the controversy of the document?
5.7 Conclusion

We challenge the two properties presented from the previous work and propose a new model that complements them. Using insights from recent social science research, we motivate and explore the first language modeling approach to detecting controversy. We find that our new approach is statistically better than prior work, while being simpler. We explore strongly controversy-indicative terms and found that a language model of documents containing “controversy” keyword directly is as helpful for this problem as complicated Wikipedia-based controversy features and more effective than existing lexicons. However, our analysis suggests that we have effectively “maxed-out” this controversy dataset. In the proposed future work, we plan to investigate the utility of CLM in terms of generalizability and explainability with the new suitable dataset.
CHAPTER 6
PROPOSED RESEARCH: TIME-ADAPTABLE CONTROVERSY DETECTION

6.1 Introduction

Most controversy detection approaches are optimized to analyze observed signals of controversial topics, such as disputes, language, or user interaction behavior on social media. However, while controversy is a dynamically changing phenomenon, the existing models do not directly capture this factor.

6.1.1 The Dynamic Nature of Controversy

Figure 6.1: Controversy computed by P score (Jang et al., 2017) among all daily tweets by date for The Dress (left), Brexit (center) and 2016 US Elections (right), reported among those Gardenhose tweets with an explicit stance. Notable peaks are annotated with associated events around that time. All dates are in UTC.

Naturally, the level of controversy changes as the topic evolves over time. For example, in a case study of controversial events, Cramer (2011) found that terms that describe *Busang case* (Depalma, 1997) have shifted from “dispute” and “controversy”
to “saga” and “scandal” over time. This demonstrates how the nature of the controversy changes as it develops. Even if the controversy does not develop further or resolves, people’s attention to the topic decreases over time, which results in lowering the level of controversy of the topic. This phenomenon is demonstrated by our study that presented a plot of the daily level of controversy measured in Twitter in Figure 6.1 (Jang et al., 2017). It shows that some controversies are more ephemeral than others. For example, “The Dress” controversy, the controversial photo that went viral when people disagreed on its colors on Twitter, was not anymore controversial on Twitter only after a few days as most people stopped caring. On the other hand, “2016 U.S. Presidential Election” had a longer span of the controversy, whose controversy has a longer-lasting effect than “the Dress.”

6.1.2 Monotonicity of Controversy Scores in Wikipedia

Existing approaches have been more focused on analyzing the controversial signals that are currently available. However, because time was not directly modeled in the existing approaches, they often have monotonic property over time. For example, M score (Yasseri et al., 2012), one of the successful measures that estimates the level of controversy in Wikipedia, is designed to be monotonically increasing over time, which limits its practical use. This is based on their algorithm design that sums over the number of mutual reverts in the edit history. The longer the edit history is, more likely we have mutual reverts, and more likely the M score gets bigger. This was demonstrated in Figure 6.2. Let us take an example of one topic that was once controversial: Michael Jackson. Figure 6.2 shows the time evolution of the M score on “Michael Jackson”. The graph shows that the controversy score has monotonically increased every time there is a new controversial event added on the article up until the point “D” where he died. However, ever since then the controversy score still remains as high as D (or higher) until later in 2012.
Some approaches are not monotonic as their scores are normalized by the number of editors who contributed to the page, which increases over time. Dori-Hacohen argued that P score (2017) can go up and down as time goes by, because they focus on the ratio of editors who are in conflict compared to the entire editor population on the topic. Their intuition is that over time if they have more editors who are not involved with disputes, the controversy score will be decreased because a lower ratio of people engage in the disputes. However, this requires more people to actively engage in non-contentious activities to cancel out the level of controversy. If simply no one cares to talk about the topic anymore, it still remains controversial over time.

6.1.3 Proposed Work: Time-window based Baselines

A straightforward solution to this issue is to compute the automated controversy scores by only considering data for a window of given time. However, we suspect that this approach will be limited by the following two challenges. First, dispute signals seem usually sparse in that they are only observed around a short period of

Figure 6.2: “Time evolution of the controversy measure of the article about Michael Jackson. A: Jackson is acquitted on all counts after five month trial. B: Jackson makes his first public appearance since the trial to accept eight records from the Guinness World Records in London, including Most Successful Entertainer of All Time. C: Jackson issues Thriller 25. D: Jackson dies in Los Angeles.” Source: http://wwm.phy.bme.hu/
time when a new controversial event triggers it as shown in Figure 6.1. When we consider a window of time, it is likely that the level of controversy is underestimated compared to real life only because we did not observe any dispute signal in the given time window, although it might exist in real life. If the topic was recently controversial but not within the given time window, it would be unrealistic to determine that it is not controversial given its history, even if we don’t observe as dispute activity on it anymore in the given time window.

Second, choosing the size of time window is a non-trivial issue as well because the characteristics of controversial topics vary. Some controversies are more long-term than the others, having more “peaks” over a long period of time. Some topics are more ephemeral than others. For example, “one-hit wonder” controversies, those that were controversial just for once (e.g., “The Dress”), are not likely to be sparked up again. Some controversies quickly go away, whereas others remain for a while. Therefore, determining the reasonable size of window to consider is not a trivial issue.

In the proposed work, we will investigate this hypothesis by analyzing a time-window-based M score.

Figure 6.3: An estimated trend line of controversy (dotted) over the observed trend (solid).
6.1.4 What happens between the controversial peaks?

We argue that in order to correctly estimate the controversy value at a given time, we need to consider the signals observed within the window of time as well as the overall history of the controversy. In this work, we assume that the dispute signals we observe through online activities are only biased samples of all controversial disputes in the real world. Our take is that while they are reliable when controversy is getting enough attention, they get underrepresented when people’s attention drops on the topic. This motivated us to ask a question: “What really happens between the controversial peaks while we don’t observe?” Therefore, we aim to estimate the controversy trend line by using the controversial peaks as reliable sample points of the real trend.

6.2 A Sketch of the Proposed Method

We plan to propose a method that estimates a dynamically-changing trend of the controversy level for a given topic. We first assume that the level of controversy naturally decays by default over time unless a new event causes another peak. We first identify “peaks” of controversy given a window of time and find a model that can best-simulate the controversy trend by connecting those peaks. We define that the model should satisfy the following three conditions:

**Condition 1:** The level of controversy for a given topic should naturally decay over time until a new event sparks another peak.

**Condition 2:** The level of controversy of topics that had more frequent or higher controversy peaks in history should decay more slowly than that of other topics that had less number of or lower peaks.

**Condition 3:** When a new controversy peak occurs, the decaying controversy trend gets reset.

From the above conditions, we present our hypothetical trend from the sample observed controversy patterns in Figure 6.3. In this figure, the left graph considers a
topic that was controversial at one point, and doesn’t have anymore peak afterwards. One of the hypothetical trend model considers an exponential decay model as shown in a dotted line. Note that the hypothetical trend implies that the topic is still controversial for a while after the peak than observed, but eventually becomes low. The graph on the right shows a topic that has three controversial peaks over time. When it had the first peak, the estimated trend follows the same decaying model from the first case given the same history so far. When it has its second peak, the trend gets reset to that peak level, and it decays more slowly than the first time. As it has more peaks, it learns that this topic has a more long-term controversy. Eventually, it decaying trend gets fit to remain somewhat controversial rather than decreasing to a zero point.

We suspect that other than its history, the decaying trend is highly correlated to people’s current attention to the topic. Therefore, we plan to utilize the statistics of how frequently the topic term has been mentioned in news articles at the time using Google Trends dataset\(^1\) for example, to fit a trend line of evolving controversy.

### 6.3 Evaluation

We plan to evaluate this work by integrating it into CLM approach for controversy detection. However, the annotated dataset does not have any time information. To correctly evaluate this method, we plan to create a new dataset including articles on the same topic but from different times over the past 10 years. We will compare against a few baselines including ones that are tailored to recognize time such as M score computed for a short time window for our purpose.

\(^1\)https://trends.google.com/trends
CHAPTER 7
EXPLAINING CONTROVERSY ON SOCIAL MEDIA

7.1 Introduction

Online controversies often emerge and evolve quickly due to the nature of social media. These platforms force users to be concise and allow them to be casual, requiring less effort to post something on Twitter than other sources, such as Wikipedia or blogs. While existing techniques enable us to identify whether a topic is controversial, understanding why it is controversial is still left as work for users. For instance, consider a following scenario: A person discovers a new hashtag movement #TakeaKnee\(^1\) on Twitter but does not know what it is about or why it is controversial at all. How would she search for people’s opinions to better understand the conflicting stances on this topic?

One straightforward approach to this problem would be for the user to search the topic and manually scan the search results until she has read enough conflicting tweets to understand the controversy. However, current search systems make this navigation difficult due to the filter bubble effect (Ingram, 2016). For example, the top posts are likely to be the ones that the user agrees with because her friends liked the posts or because she or her friends follow the authors.

Another strategy for navigating Twitter is to identify a few key hashtags that indicate stances and then search for posts that contain them. As people are forced to write posts under the strict character limit, certain hashtags are utilized as self-created labels for their opinions (e.g., #imwithher in support of Hillary Clinton

---

\(^1\)This was prevalent during the US national anthem protests that began in 2017.
or #MAGA in support of Donald Trump during the 2016 US presidential election). However, because the use of hashtags (even the ones that seemingly contain obvious stances) are known to be noisy (Mohammad et al., 2016a), the user must still carefully read through each tweet. More importantly, she has to go through a large number of noisy tweets that are not useful to understand the controversy while using her own judgment to identify their stance (if they even have one). This process requires substantial effort, critical reasoning, and phenomenal patience. It is clear that users could benefit from automating this process.

We propose a technique that generates a stance-aware summary by selecting the top tweets that best explains a given controversy.

Table 7.1: An example of good (top) and bad (bottom) summary tweets on “Abortion” posted on Nov 4, 2016. The good summaries are selected from our method. Examples of stance hashtags are marked in bold.

<table>
<thead>
<tr>
<th>Good Summaries</th>
<th>Bad Summaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>• We know it’s not okay that for 40 yrs politicians have denied a woman coverage of abortion just because she’s poor #BoldTheVote #BeBoldEndHyde</td>
<td>• lmaaoaoao b**** i would did the abortion myself right there lmaaoaoao</td>
</tr>
<tr>
<td>• Read the whole story about #HarvardSoccer before forming idiotic tweets. Don’t support #RapeCulture by calling it #LockerroomTalk</td>
<td>• before I formed you in the womb I knew you jer 1:5#prolife</td>
</tr>
<tr>
<td>• Hillary Clinton voted no to banning late-term abortions, even though over 80% of Americans support the ban. #VoteProlife</td>
<td>#Defundpp [URL] #UnbornLivesMatter</td>
</tr>
<tr>
<td>• Abortions: the new fall trend in religious circles [URL]</td>
<td>• Could you imagine crying over ur uni stopping anti abortion protests, if you’re so pro life then go and f***ing get one?</td>
</tr>
</tbody>
</table>

7.2 Approach

We first discuss what makes a tweet a good summary. We then develop a ranking model that ranks the tweets by how likely a tweet is part of a good summary. Finally, we propose two methods to select the summary from the ranked tweets.
7.2.1 Ranking Model

Based on the definition of controversy by previous work, we define a good controversy summary as a description that effectively captures different arguments of two communities that take conflicting stances with each other. After examining many examples (see Table 7.1), we derive three primary components that characterize a good controversy summary tweet:

- **Stance-indicative (S):** A good tweet strongly indicates its stance and is often followed by some particular stance hashtags that are widely used by users from the same stance community. While both good and bad tweets frequently include stance hashtags, the presence of stance hashtags is a positive reinforcement signal if the quality of tweet is decent.

- **Articulation (A):** A good tweet is clear, persuasive, and logical. It is also written with proper language.

- **Topic Relevance (T):** A good tweet is relevant and self-explanatory in the context of a particular topic.

For any controversial topic \( T \), we assume that there are always two stances that are in conflict with each other. We denote these stances as \( S_A \) and \( S_B \). Let \( \Gamma \) be a summary of a given topic \( T \). We let \( \Gamma = [\Gamma_A, \Gamma_B] \) that denotes the summary of \( S_A \) and \( S_B \), respectively. We define a model that computes whether a tweet \( \tau \) is likely to be in the set \( \Gamma_A \):

\[
P(\Gamma_A|\tau) = f(P_S(S_A|\tau), P_A(\tau), P_T(\tau|T))
\]  

(7.1)

where \( P_S(S_A|\tau) \) computes how likely a tweet indicates \( S_A \), \( P_A(\tau) \) computes how articulate the tweet is, and \( P_T(\tau|T) \) computes how relevant the tweet is for the topic.

In the next sections, we discuss how to estimate the first two scores. For the topic relevance score, we use the straightforward probability that the tweet sentence
was generated from the language model of the given topic, normalized by the tweet length.

### 7.2.2 Estimating Stance-indication

To estimate stance-indication, we first identify stance hashtags that statistically characterize the stance community. We use the stance hashtags as a proxy to estimate the tweets that indicate the same stance as follows:

\[
P_S(S_A|\tau) = \sum_{h \in H} P(h|\tau) \cdot P_S(S_A|h) \cdot P(h)
\]

Then the score boils down to estimating \( P(h|\tau) \), a probability that the tweet includes a given hashtag \( h \), and \( P_S(S_A|h) \), a score that indicates how likely \( h \) represents \( S_A \). As \( S_A \) and \( S_B \) are mutually exclusive, we penalize ambiguous tweets that are likely to contain stance hashtags of the opposing side by subtracting the score for the opposite stance as follows:

\[
P_S(S_A|\tau) = \sum_{h \in H_A} \left[ P(h|\tau) \cdot P_S(S_A|h) \right] - \sum_{h \in H_B} \left[ P(h|\tau) \cdot P_S(S_B|h) \right]
\]

where \( H_A \) and \( H_B \) are the set of stance hashtags that represent \( S_A \) and \( S_B \) respectively.

#### 7.2.2.1 Identifying Stance Hashtags \((H_A, H_B)\)

To obtain a set of stance hashtags, we first identify two communities, \( C_A \) and \( C_B \), each of which represents two conflicting stances, \( S_A \) and \( S_B \). As introduced by Garimella et al., we construct a user retweet (RT) graph and partition it into two groups (\(?\)). We use a simple method that produces only two communities so as not to deal with the extra step of classifying several identified communities to two stances. We leave identifying multiple communities and clustering them into one of the stances of interests to generate the summaries from for the future work.
Once we identify $C_A$ and $C_B$, we assume that tweets that are written by users from $C_A$ and $C_B$ are likely to indicate $S_A$ and $S_B$ respectively. From the two sets of tweets, we compute the information gain (Yang and Pedersen, 1997) that each hashtag gets for the information of the community class when they are present in the tweets: if we know nothing about the tweet but the hashtag presence, which hashtag best indicates its stance community? Finally, we define $\mathcal{H}_A$, the set of stance hashtag of $S_A$, as follows.

$$\mathcal{H}_A = \{h \in \mathcal{H} | h \in \text{TopN}(IG) \land freq_A(h) > freq_B(h)\}$$

where $IG$ is a function that returns the information gain value for the two stance classes for a given hashtag, $freq_A$ is the frequency of $h$ in the tweets published from $C_A$, and $\text{TopN}(IG, \mathcal{H})$ returns the $N$ items that have the highest scores from a given function $IG$ among the items in the given set $\mathcal{H}$. In our experiments, we set $n = 30$, which covers a sufficiently high number of tweets in the community given that the distribution of hashtag frequency follows the power law (Pérez-Melián et al., 2017). We then let $P_S(S_A|h)$ be the normalized score of $IG(h)$ for all hashtags in the set $\mathcal{H}_A$.

### 7.2.2.2 Estimating $P(h|\tau)$ via Latent Hashtags

If we think of hashtags as user-generated annotations, hashtags are incomplete annotations. It means that a lack of a certain hashtag does not necessarily mean that it is not a relevant label. To better utilize hashtags as more accurate signals, we make hashtags more complete annotations by estimating $P(h|\tau)$ for all hashtags, the probability that tweet $\tau$ generates a hashtag $h$. Therefore, we adopt a character composition model, TWEET2VEC, which finds a vector space representation of tweets to predict user-annotated hashtags (Dhingra et al., 2016).

By finding the embeddings of tweets and hashtags, we estimate $P(h|\tau)$ for hashtags that were not explicitly used in the given tweet. The model computes the hashtag
Table 7.2: The features used to train a regression model for predicting the level of
tweet articulation.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tweet POS Tags (Owoputi et al., 2013)</td>
<td>The ratio of Tweet POS tags</td>
</tr>
<tr>
<td>OOV words ²</td>
<td>The ratio of words that are not in the dictionary</td>
</tr>
<tr>
<td>Offensive Words ³</td>
<td>The ratio of offensive/profane words</td>
</tr>
<tr>
<td>POS Tags N-grams</td>
<td>N-grams of Tweet POS Tag sequence</td>
</tr>
<tr>
<td>Stop words</td>
<td>The ratio of stop words</td>
</tr>
<tr>
<td>Tweet length</td>
<td>The number of characters in a tweet</td>
</tr>
<tr>
<td>Avg. word length</td>
<td>The avg. number of characters in tweet words</td>
</tr>
</tbody>
</table>

posterior probability for a given tweet for all hashtags in their softmax layer in order
to find the top hashtag predictions. We use this probability as \( P(h|\tau) \) for hashtags that were not explicitly used in the given tweet.

7.2.3 Estimating the level of articulation

We build a regression model that predicts how well the tweet is written and
generate an annotated set of 150 articulate and 150 non-articulate tweets on arbitrary
topics. The annotation criteria between the two classes is whether the given tweet is
logical, the grammar is sound, and it is written with proper language.

Similarly, Duan et al. propose a classifier to evaluate the content quality of tweets
(Duan et al., 2012). In addition to their features, we include a large set of POS
tags that are Twitter-specific provided by TweeboParser (Owoputi et al., 2013), N-
grams of the POS tags sequence to capture the structural flow of the good sentences,
and the ratio of offensive words to penalize usage of inappropriate language. This
model is generalizable since the features are not content-specific. We trained a lo-
gistic regression model and obtained 89.9% classification accuracy using 5-fold cross
validation.

²http://wordlist.aspell.net/12dicts

³https://www.cs.cmu.edu/~biglou/resources/bad-words.txt
7.2.4 Summary Selection

We propose two algorithms that aggregate the three probability scores to generate the final $k$ summary, which we set as 10 in our experiments. To produce a final summary to equally cover two stances, both algorithms select $k/2$ tweets from each stance.

SUMSAT ranks the tweets by setting the aggregation function $f$ (in Eq. 7.1) to be a harmonic mean for the three scores described earlier. HASHTAGSUMSAT, on the other hand, while using the same aggregation function, first identifies the top $k/2$ stance hashtags for each stance and selects the top tweet for each hashtag. While we use a harmonic mean as $f$, any aggregator can be plugged in. The difference of the two algorithms come from whether it globally ranks the tweets or ranks the tweets per each hashtag.

7.3 Evaluation

We evaluate our methods by running them on real data and conducting user studies to capture the utility of our algorithms.

7.3.1 Experiment Setup

We consider five controversial topics including two short-term, event-based controversies (2016 US Presidential Election and 2017 US National Anthem Protests which we refer to as #TakeAKnee), and three long-term ethics-related controversies (Abortion, Feminism, and Climate Change).

Our goal is to generate a summary that can explain why the topic is controversial. For each topic, we generate a pair of summaries and ask 10 participants on Amazon Mechanical Turk which summary better explains the controversy in a double-blind fashion. A pair of summaries were compared twice by two participants. The participants could also say that the quality of the two summaries is the same. To observe
Table 7.3: The amount of data used to train Tweet2Vec and summary generation. The number in parentheses refers to the number of tweets published by the stance community.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Tweet2Vec</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Tweets</td>
<td># Users</td>
</tr>
<tr>
<td>Election</td>
<td>10.8M</td>
<td>4.3M</td>
</tr>
<tr>
<td>#TakeAKnee</td>
<td>565K</td>
<td>692K</td>
</tr>
<tr>
<td>Abortion</td>
<td>692K</td>
<td>539K</td>
</tr>
<tr>
<td>Feminism</td>
<td>1.7M</td>
<td>1.7M</td>
</tr>
<tr>
<td>Climate Change</td>
<td>546K</td>
<td>360K</td>
</tr>
</tbody>
</table>

whether a subset of tweets whose author’s stance is identified from the community generates a better quality summary, we experiment with two cases for each algorithm: (1) using all tweets as summary candidates or (2) using only tweets whose author belongs to one of two stance communities we identified. We distinguish the second case by adding ‘C’ (for the community) to the method name. We also generate summaries including the following baseline methods:

- **Random**: A random set of $k$ tweets from a unique set of tweets.
- **MostRT**: The top $k$ most-retweeted tweets in a given day
- **SumBasic** (*Nenkova and Vanderwende, 2005*): A general summarization technique. We preprocess the tweets to exclude Twitter-specific stop words.

### 7.3.2 Results and Discussion

The evaluation shows that our methods were consistently more effective than other baselines across all five topics as shown in Figure 7.1). Overall, SumSAT generated the summaries that were preferred the most (68%) followed by HashtagSumSAT-C (61%). We report the results by the five topics in Figure 7.2.
Figure 7.1: The evaluation results by the methods. The rightmost four bars are our methods.

Figure 7.2: The user study results by the topics. The rightmost four bars in each topic are our methods. We did not include SumBasic in the graph because it was the worst method for all topics, being preferred only 8% of times overall.
Controversy summarization as a new task: Overall, both Sumbasic (8%) and Sumbasic-C (42%) generated the worse summary than the naive baselines such as mostRT or random. This suggests that controversy summarization is an inherently different task from a general topic summarization.

MostRT is often a strong baseline, but its performance is not reliable: For the topic of #TakeAKnee, the mostRT baseline was as effective as our top approach. The topic also particularly had a high ratio of retweets compared to other topics (Table 7.3). However, depending on the topic and the day, mostRT can also be the worst feature, even worse than random selection as in the case for the topic of Feminism. For example, the top retweets in Feminism include ‘Happy International Women’s Day!’. Retweets can often be tweets for entertainment and can easily be dominated by people on one side of stances who are more vocal on Twitter.

Social features seem to be more useful than the content itself in stance summarization: We learned that in identifying and finding stance-indicative tweets, social features are far more important than the content itself. For example, mostRT outperforms a general summarization technique that only considers the text content most of the times. This finding aligns with the findings of the previous study on detecting controversy on Twitter (Garimella et al., 2016).

Utility of stance hashtags: While SUMSAT was an overall winner, HASHTAG-SUMSAT outperformed SUMSAT for two topics: US Election and #TakeAKnee. We observe a tendency in the event-based controversies like those topics show more active usage of stance hashtags as there were specific actions people try to promote via stance hashtags. In such type of controversies, stance hashtags were particularly effective to generate a summary around.
7.4 Conclusion and Future Work

We introduce and tackle a new task of generating a stance-aware summary to explain controversy on social media. Our goal is to provide a tool that helps people navigate controversy effectively. We propose a ranking model that considers three factors that suggest a tweet be part of a good summary derived from our qualitative observations. We assume that a good summary tweet is clear, articulate, and relevant to the topic. Our algorithm characterizes two conflicting stances by identifying two communities from a retweet graph and retrieving the tweets published by them. We define and identify “stance hashtags” that are distinctively used to indicate their opinions in each community and propose a probability model that computes how a tweet is likely to indicate the stance of the community based on the probability that the tweet is likely to generate those hashtags. Our evaluation demonstrates that users prefer the summaries from our methods over the ones from other reasonable baselines.

7.5 Proposed Work: More Rigorous Evaluation

We plan to discuss about more rigorous evaluations including IRB approval if necessary and collecting more answers to ensure that the results are statistically significant. We also acknowledge that this evaluation method is not scalable because it is expensive and slow. In addition, simply by asking the overall quality of the summary of the results, it is hard to obtain a deeper analysis on the quality of the summaries in depth, such as in which aspect one summary was better than the other. Hence, we will investigate an intrinsic evaluation method that is similar to traditional recall and precision in a controversy summary context. This will lay a solid path for future endeavors towards the problem, while allowing us to further understand the nature of the problem by studying the aspects that a controversy summary should satisfy.
CHAPTER 8
RESEARCH PLAN

In this last part of the proposal, we outline the research plan for the proposed work proposed in Chapter 1.1 and other dissertation related work as follows.

- **May 2018**: Thesis Proposal
- **June 2018**: Regarding time-adaptive controversy model, I will first implement the time-split M score (Yasseri et al., 2012) as the state-of-the-art baseline. For a few controversial topics that have been dynamically changed, I start by demonstrating how the state-of-the-art technique fails to adapt.
- **July 2018**: I will develop a model for time-adaptive controversy model and perform an evaluation. I will also conduct experiments to investigate generalizability and explainability for controversy language model.
- **Aug 2018**: I will wrap up the time-adaptive controversy model research and prepare a paper submission for ECIR 2019 or WSDM 2019.
- **Sep 2018**: I will work on more rigorous evaluation framework for controversy summarization work.
- **Oct 2018**: I finish all the remaining necessary experiments and start working on the dissertation.
- **Nov 2018**: The dissertation preparation
- **Dec 2018**: Ph.D. dissertation defense


Shiri Dori-Hacohen, Elad Yom-Tov, and James Allan. Navigating controversy as a complex search task. In *SCST@ECIR*, 2015.


Kristen A Johnson and Dan Goldwasser. Identifying stance by analyzing political discourse on twitter. 2016.


