INTERACTIVE TEXT ANALYTICS FOR USER-GENERATED CONTENT

by

Raheleh Makki Niri

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
April 2017

To my family,

where love never ends.

Table of Contents

List of Tables . . . . . **vi**

List of Figures . . . . . **viii**

Abstract . . . . . **x**

List of Abbreviations Used . . . . . **xi**

Acknowledgements . . . . . **xii**

Chapter 1 Introduction **1**

1.1 Motivation 1

1.2 General Background 3
1.2.1 Active Learning 4
1.2.2 Visualization and User Interaction 4

1.3 Goals and Objectives 6

1.4 Outline 8

Chapter 2 Context-Specific Sentiment Lexicon Construction for Sentiment Analysis **10**

2.1 Introduction 10
2.1.1 Research Problem 11
2.1.2 Overview 12

2.2 Related Work 13
2.2.1 Generating Sentiment Lexicons 13
2.2.2 Visualizing Sentiment Words 16
2.2.3 Sentiment Analysis 19

2.3 Proposed Method 21
2.3.1 Automatic Lexicon Creation 22
2.3.2 Visualization 26

2.4 Evaluation 35
2.4.1 Dataset 35
2.4.2 Alternative Interface 36
2.4.3 User Study 37
2.4.4 Results 38

2.5 Conclusions . . . 46
2.5.1 Limitations and Future Work . . . 47

Chapter 3 Microblog Retrieval for Parliamentary Debates . . . 50

3.1 Introduction . . . 50
3.1.1 Research Problem . . . 52
3.1.2 Overview . . . 53

3.2 Related Work . . . 54
3.2.1 Processing Microblog Content . . . 54
3.2.2 Visual Analytics for Microblog Content . . . 58

3.3 Proposed Method . . . 59
3.3.1 Unsupervised Tweet Retrieval . . . 62
3.3.2 Active Tweet Retrieval . . . 63
3.3.3 Interactive Visualizations . . . 69

3.4 Evaluation . . . 74
3.4.1 Datasets . . . 74
3.4.2 Parameter Setting . . . 77
3.4.3 Retrieval results . . . 78
3.4.4 Use Cases . . . 83
3.4.5 ATR-Vis Pair Analytics Evaluation . . . 92
3.4.6 Discussion . . . 97

3.5 Conclusions . . . 101
3.5.1 Limitations and Future Work . . . 102

Chapter 4 Microblog Filtering based on User Interest Profiles . . 105

4.1 Introduction . . . 105
4.1.1 Research Problem . . . 106
4.1.2 Overview . . . 107

4.2 Related Work . . . 108

4.3 Proposed Method . . . 113
4.3.1 Unsupervised Tweet Filtering . . . 113
4.3.2 Unsupervised Query Expansion by Semantic Relatedness Methods . . . 121
4.3.3 Active Query Expansion by Semantic Relatedness Methods . . . 122

4.4 Evaluation . . . 128
4.4.1 Dataset . . . 129
4.4.2 Evaluation Metrics . . . 129

4.4.3 Threshold Removal for Non-Silent Days . . . 131
4.4.4 Results: UTIF and Automatic Query Expansion . . . 132
4.4.5 Results: ACTIF and Active Query Expansion . . . 134
4.4.6 Discussion . . . 141

4.5 Conclusions . . . 143
4.5.1 Limitations and Future Work . . . 144

Chapter 5 Conclusions . . . 148

5.1 Future Research Directions . . . 150

Bibliography . . . 153

Appendix A Screening Questionnaire . . . 181

Appendix B Demographic Questionnaire . . . 182

Appendix C Post-study Questionnaire . . . 184

Appendix D Dalhousie Ethic Board's Letter of Approval . . . 186

Appendix E Tweet Fields . . . 188

Appendix F Tweet Distribution Over Time . . . 189

Appendix G Evaluation Metrics . . . 190

Appendix H Applying SVMRank for Tweet Filtering . . . 191

Appendix I Applying RoundRobin to UTIF . . . 192

Appendix J Performance of Tweet Filtering Systems . . . 193

List of Tables

1.1	Terminology used in this thesis	5
2.1	Notation used in this chapter	21
2.2	Dependency relations used for extracting sentiment pairs	24
2.3	User study tasks and datasets	38
2.4	Average, standard deviation, and the χ^2 test of the sentiment lexicon accuracy before and after user supervision using text-based and visual interfaces	39
2.5	ANOVA table for the accuracy of the first task	40
2.6	ANOVA table for the accuracy of the second task	40
2.7	Instructions for each task and their relative time length	42
2.8	The average and standard deviation of the amount of time spent by the participants for each dataset (in seconds)	42
2.9	ANOVA table for the amount of time spent by the participants for the first task (in seconds)	43
2.10	ANOVA table for the amount of time spent by the participants for the second task (in seconds)	43
3.1	Notation used in this chapter	61
3.2	Statistics of collected Canadian and Brazilian Twitter datasets	76
3.3	Results of the unsupervised retrieval method, a random active retrieval, ReQ-ReC, and the ATR-Vis' selection strategies	79
3.4	Results obtained with the unsupervised retrieval method and the ATR method, for each debate in the Canadian dataset	81
3.5	Accuracy, macro-precision, macro-recall, R-precision and MAP for the unsupervised retrieval method, and the ATR-Vis' selection strategies using the Brazilian dataset.	82
3.6	Results obtained with the unsupervised retrieval method and the ATR-Vis' selection strategies, for each debate in the Brazilian dataset	83

4.1 Notation used in this chapter 114

4.2 Results when combining UTIF with different strategies 132

4.3 Sample tweets returned/missed by UTIF and its combinations for profile "Mr. Holmes Movie" considering word2vec embeddings in query expansion and $\theta = 1$ 133

4.4 Results of UTIF+NEI and top 4 TRECMicroB participants . . 134

4.5 Percentage of relevant tweets in the collected dataset, in the union of all matched tweets, and in the top two ranked tweets by the language model 135

4.6 SVMRank results with different sampling strategies considering 1020 labeling requests 136

4.7 Summary of different methods 139

4.8 nDCG and MAP for ACTIF, SVMRank, UTIF+NEI and Round-Robin 139

4.9 Results of the semantic text relatedness methods based on word embeddings using nDCG-1@10 142

J.1 Profiles and their nDCG-0@10 value when UTIF, ACTIF and SVMRank are applied considering 1020 labeling requests 193

List of Figures

1.1 Proposed generic framework for interactive text analytics of user-generated data 7

2.1 Proposed approach for constructing context-specific sentiment lexicon 12

2.2 Rose plot showing different affects and a petal presenting the range of affect 18

2.3 SocialHelix: a) mimicking the structure of DNA molecule, b) showing temporal transition of sentiments, c) the event view . 18

2.4 Extracting sentiment pairs from the set of reviews 23

2.5 An overview of the Neighbor Joining algorithm for constructing the tree from the distance matrix. 28

2.6 Tree cloud view for a printer review dataset 29

2.7 Polarity assignment view for a printer review dataset 31

2.8 Polarity assignment view after the user moved the sentiment word "cheap" and duplicated the sentiment word "low" by clicking on its aspect "quality". 32

2.9 Polarity assignment view after the corrections were made by the user 34

2.10 Text-based interface for constructing context-specific sentiment lexicon 36

2.11 Stacked bar charts showing the frequency of the participants' answers to a number of questions. 44

2.12 Concept map resulting from qualitative analysis of open text answers 45

3.1 Proposed framework for retrieving tweets relevant to a set of political debates 60

3.2 Hashtag selection strategy to improve retrieval precision and recall 65

3.3 Assignment View: visual aids to facilitate tweet retrieval . . . 71

3.4 More View: a set of visual aids to facilitate tweet retrieval . . 75

3.5 Canadian dataset: for each selection strategy, the number of correctly retrieved tweets 80

3.6 Ring visualization: identifying weak connections between unretrieved tweets and under-represented debates. Colors indicate different debates and two tweets are connected if their similarity score is above a user-selectable threshold for one common debate. 85

3.7 Beginning of the user involvement considering the Brazilian parliamentary dataset 87

3.8 After a batch of user feedback using the Brazilian parliamentary dataset 88

3.9 The Assignment View showing the labeling request, discriminative features, and keyword distributions for tweets on emerging news stories. 90

3.10 The Force Layout shows the tight connection between the story of "Dallas Shooting" (in orange) and "Killing of Afro-Americans" (in pink). 90

3.11 ATR-Vis can be used to improve the recall of a story 91

3.12 Precision for each debate before and after adding more tweets to the evaluation set 99

3.13 Accuracy before and after applying ATR strategies with different values for the number of keyterms 99

4.1 Proposed framework for filtering tweets based on user interest profiles 115

4.2 Proposed active query expansion strategies for tweet filtering based on user interest profiles 126

4.3 nDCG@10 of ACTIF, SVMRank, and Round-Robin for different number of labeling requests 140

4.4 Results of UTIF and its combinations with WikiSim for different values of θ 141

F.1 Distribution of relevant tweets over time 189

Abstract

The rapid growth of social media platforms, weblogs and online forums has made the volume of user-generated content increase exponentially in recent years. User-generated content is different from traditional documents in structure, length, and semantics. Consequently, applying traditional natural language processing and text mining methods to emerging and challenging text mining problems does not always achieve satisfactory results. In other words, as data changes, their characteristics and features change, and therefore the solutions that rely on certain assumptions about the data, which may no longer be valid, fail to perform as expected. In addition, the users' information needs may change over time, and hence are the type of applications that provide answers to these needs.

This thesis studies the impact of actively involving the user in the analytical process of such data on overcoming related challenges and improving the quality of the analysis. We investigate whether employing active learning and visualization techniques increases the benefits gained from incorporating user knowledge, and whether these techniques enhance user involvement. Moreover, our ultimate objective is to assist users to better understand the data and make decisions. We evaluate this approach considering different online applications and datasets. First, we develop and evaluate solutions for the problem of sentiment classification of context-specific opinion words in product reviews, with a focus on minimizing user effort using visualization techniques. Second, we address the problem of topical classification of microblog posts by introducing active learning and visualization techniques to augment user engagement. The third part of our research addresses the problem of Twitter information filtering based on user interest profiles. We propose active learning techniques with a focus on query expansion. For all these cases, our results demonstrate that incorporating user knowledge improves the performance of automatic methods significantly, and using active learning and visualization techniques for tailoring user engagement methods increases the gain obtained from user supervision.

List of Abbreviations Used

ANOVA	Analysis of Variance.
MAP	Mean Average Precision.
NDCG	Normalized Discounted Cumulative Gain.
NLP	Natural Language Processing.
SVM	Support Vector Machine.
Tf-Idf	Term frequency, Inverted document frequency.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Evangelos E. Milios and Professor Stephen Brooks, for their continuous support during my Ph.D study, and for their patience, motivation and guidance. I am ever grateful for the opportunity to work with them and to learn from them.

My sincere thanks and appreciation goes to Dr. Axel J. Soto. The completion of my work would not have been possible if not for his generous support and valuable feedback and guidance. Thank you for your part in my journey.

I would like to greatly thank Professor Maria Cristina Ferreira De Oliveira and Professor Rosane Minghim for their encouragement and constructive ideas, and Eder Carvalho for developing the visualization component of ATR-Vis in Chapter 3 and collaborating in the process of system evaluation.

Also, special thanks to the rest of my thesis committee: Professor Vlado Keselj, Professor Kirstie Hawkey and Professor Stan Matwin, who have given their time and expertise generously to better my work. Special thanks to Professor Julita Vassileva from University of Saskatchewan, for kindly accepting to be my thesis external examiner, and for her insightful questions and suggestions that certainly improves my thesis.

My deep appreciation to my colleagues Armin Sajadi and Magdalena Jankowska for the inspiring discussions, thoughtful support, and for all the great times that we have had in the last five years. Special thanks to all members of MALNIS research group for their valuable feedback.

Last but not the least, I would like to thank my family, especially my parents Robabeh and Ahmad, my husband, my brother and my sister for always being there for me. Anything good that has come to my life has been because of your example and love. Your prayers for me was what strengthened me so far. I will never be able to thank you enough.

Chapter 1

Introduction

1.1 Motivation

The amount of unstructured textual data being produced is increasing markedly. Emails, blogs, tweets, and customer reviews generate online textual data continuously. In addition, it is crucial for businesses such as news media, data analytics and policy makers to summarize, understand and extract meaningful information out of this sheer volume of data to make better decisions. For instance, marketers need to explore and analyze customer reviews to understand customers' interests, discover patterns, and gain deeper insights in order to invest in areas that leads to higher profits and user satisfaction.

Online textual content is different from traditional documents with regard to its structure, vocabulary, length and semantics. In addition, user-generated content are often temporally sequenced. For instance, social media posts, comments on discussion forums, and comments on content-sharing websites such as YouTube, are chronologically ordered. With these differences, the performance of text analytics methods that are designed, implemented for, and evaluated on collections of static corpora, may deteriorate when applied to noisy and unstructured user-generated content. Although various methods have been introduced by researchers to extract high-quality information from unstructured text, there are still several challenges to be overcome.

One of the main challenges of processing textual data is that natural languages are ambiguous at different levels of processing [132], which makes understanding unstructured textual data difficult. This challenge becomes more severe for the data with problematic characteristics. For instance, online data such as chats, tweets, forums and blogs contain spelling errors, incorrect grammar, acronyms and non-standard words. Therefore, the performance of natural language processing techniques, with satisfactory results on formal text, deteriorates when applied to user-generated text.

For instance, it has been shown that the performance of standard part-of-speech taggers [91] and named entity recognizers [233, 57] degrade significantly when they are applied to Twitter data.

Moreover, the acquisition of sufficiently large and representative labeled datasets is a difficult task in most text analysis applications. Labeled data is required by supervised techniques to discover the relationships among the predictors and the target variable in order to construct a model that can be applied to unlabeled data. This data is also needed for evaluating trained algorithms regardless of the learning techniques that are used. However, in most cases labeled data is not sufficiently available and manually constructing it is a tedious and time consuming task. This problem is more critical for high-velocity and varied user-generated data, such as social media platforms and discussion forums, where the labeled data becomes quickly outdated [77, 239, 31]. For instance, as Twitter users discuss a broad range of new emerging topics every day that are based on real-time events happening in the world, learned topic extraction models on previous days' posts may not be successful in topic detection of users' new tweets [281]. Therefore, it is not efficient to generalize observed patterns from past datasets to the unseen data. In other words, in a dynamic real-world setting, where the data changes over time, the characteristics of new data often diverge from the training data. Therefore, with the domain variation and requirements specific to different applications, it will not be possible to effectively use the same learned algorithm for every task and/or dataset.

The ever growing popularity of social networks, discussion forums, and weblogs has not only led to an increase of user-generated content on these platforms, but also to the number of online users, each with their specific needs of finding other users' shared information (news, opinions, thoughts, or personal experiences). The large amount of information presented to users in these platforms results in information overload [21, 235], where users cannot identify most relevant information to their specific needs. In addition, users with different backgrounds and interests seek different results from the same application. For instance, two users may consider different documents to be relevant to the same query in an information retrieval task [263]. This increase in the number and variety of information needs makes tasks of extracting, categorizing and summarizing useful information from the large volume of unstructured and noisy

user-generated content more challenging.

These persistent challenges mean that while text analytics techniques have been increasing their capacity for understanding and modeling data, their performance on user-generated content should be further improved to meet users' expectations. The fact that users' domain knowledge can help improve the performance of the learned algorithm from datasets [190], motivated us to leverage users' analytical abilities in challenging tasks of processing user-generated content. There is a significant opportunity to improve the results by enabling the user to evaluate the trained model and provide input about the data based on her expert opinion. In this way, the model can be modified to better fit the user needs. Recently, different visualization [255], user interaction [151], and active learning techniques [50, 274] have been married with text analytics methods aiming at gaining the most from user feedback while minimizing the supervision effort.

Furthermore, there are many researches on demonstrating the importance of analyzing user-generated data and making strategic decisions in the success of businesses [7, 121]. As collections of the data get bigger and more diverse, the task of locating information and making decisions become more difficult, costly and time-consuming for domain experts and data analysts. In these cases, the user needs to perform explanatory analysis in order to understand the data and make better decisions. Consequently, it is crucial to develop text analytics tools that help users explore, analyze and understand data in a better, faster and easier way [285, 46].

Therefore, there are two types of users: 1) users of online systems or services who generate online content, such as authors of tweets, product reviewers, and/or bloggers; 2) users who are involved in and benefit from the analysis process of data, whose knowledge is incorporated in the process through provided feedback. In this thesis, where we discuss user involvement and user supervision, we refer to the latter group of users, who are interested in exploring and analyzing user-generated content (e.g. data analysts, journalists, product managers, politicians).

1.2 General Background

A general background of the techniques we use in this thesis is provided in this section. We first discuss active learning techniques, which are a special case of semi-supervised

methods, and then a brief overview on user interaction techniques that are employed when human users are involved.

1.2.1 Active Learning

Semi-supervised learning techniques make use of large amounts of unlabeled data along with a small amount of labeled data to build better learners. These techniques intend to understand how combining labeled and unlabeled data improves learning [311]. This can be of great value when the acquisition of fully labeled training sets is impractical.

A special case of semi-supervised learning techniques apply strategies in which the learner interactively chooses the data from which it learns. Active learning refers to these techniques where the learner asks an information source about the label of an unlabeled data instance. In general, the more labeled data are available, the more knowledge the learner has access to and the better they perform. However, the usefulness of each data instance varies as they contain different information. Hence, selecting the instances to be labeled by the user is crucial for increasing the gain obtained from user supervision. The idea behind active learning is that if the learning algorithm controls the instances that are asked from the information source, it will achieve higher accuracy with less training data [206]. One common approach is to select the most informative and representative instances to be labeled [120]. We refer to the selected instances to be labeled as labeling requests. The terminology we use in this thesis is defined in Table 1.1.

1.2.2 Visualization and User Interaction

When the human is involved in the learning process, visualization and human computer interaction techniques are sometimes applied to facilitate user supervision. Interactive visualizations have been widely used in the area of visual analytics to help users better observe and explore the data space, understand the data, and reveal hidden relations [264]. The visual interfaces can also accelerate the acquisition of high level information from the data, which helps with making decisions and modifying the data or the model for improving the quality of the results [140]. Many studies have been devoted to the analysis and exploration of text using visual analytics [9]. Visual

Table 1.1: Terminology used in this thesis

Term	Definition
Information Source	A human user or a simulated user that provides information about instances a method requests for labeling
Automatic Method[1]	An unsupervised, semi-supervised or supervised method that performs a given task without any involvement or supervision from an information source
Active Method	A method that performs a given task by interactively asking an information source to label the instances that are selected by the method itself
Labeling Requests	Instances that are selected by active methods to be labeled by an information source[2]
Selection Strategy	Strategies that determine which data instances should be selected as labeling requests to be asked from the information source

[1]It is also known as "passive method" in literature.

[2]In literature, the term "query" is often used instead of "labeling request". However, in this thesis, we use "query" to refer to information needs that users submit to a search engine or information filtering system.

analytics aim to combine the benefits of data mining methods with human users' cognitive abilities and domain knowledge to perform analytical tasks that cannot be automated [139].

User involvement in interactive visualizations can be categorized into four different groups [197] based on two dimensions, the direction of information and the entity of interest. The information is either passed from the algorithm to the user, or from the user to the algorithm. The former and latter directions are introduced as feedback and control by Mühlbacher et al. [197], respectively. The entity of interest consists of execution and results. Execution is the information about the computation of the algorithm, while results indicates the information about final or intermediate results. Therefore, the four user involvement categories are execution feedback (e.g. showing the computation progress to the user), result feedback (e.g. showing the intermediate results of the computation), execution control (e.g. user canceling the computation) and result control (e.g. user interacting with the computation in order to steer the results). Result control is the most common type of user interaction in the visual analytics literature. Although our interactive systems may contain all types of user involvement, result control and feedback are the main types we consider in this thesis. It is important to note that throughout this document, "user feedback" refers to the information the user provides for the system, which is different than the meaning of feedback in these categories. We also note that this categorization of input and feedback could apply to systems with interaction ranging from simple text interaction all the way to complex multi-view visualizations.

Traditionally, designers and developers would meet with domain experts after creating a system to receive feedback about the quality of the generated results. Then, developers would adjust the system to address feedback from domain experts and would meet with them again. This asynchronous iterative process was lengthy and laborious [10], which has led researchers to develop interactive systems. In interactive systems, learning iterations and model updates are more rapid and incremental [10]. In other words, the system is updated immediately after the user input and often the magnitude of the changes are small, which enables users (even non-technical users) to examine the impact of their input on the results and adapt their subsequent interactions until desired outputs are observed.

The role and importance of users within the interactive systems have been demonstrated through several case studies [10]. It has been illustrated that when systems are transparent about their learning algorithm, users may have a better understanding of the process, which in turn enhances their experience and helps them provide better feedback that improves the performance of interactive systems. For instance, it has been shown that the more users understand the reasoning of a recommender system, the more and better feedback they provide and the more satisfied they are with the received recommendations [148]. In addition, users are inclined to provide more than just data labels and would like to have more control over the interactions [10]. However, since providing transparency of the systems and giving users more control may lead to cognitive overload, further studies are needed to investigate their effects on the performance of the interactive systems in different scenarios.

This thesis presents interactive systems for several important and related tasks involving the processing of user-generated content. We believe that studying how people interact with these systems and proposing new techniques for improving the interaction can result in better user experience as well as more effective systems.

1.3 Goals and Objectives

The goal of this thesis is threefold: 1) to involve the user in the analysis of user-generated content to enhance the quality of the results by the automatic methods, 2) to design, develop, and analyze techniques that make user supervision easier, more

efficient, and less expensive, and 3) to assist the user in the sense-making of user-generated content by employing active learning, visualization and interactive analytics techniques.

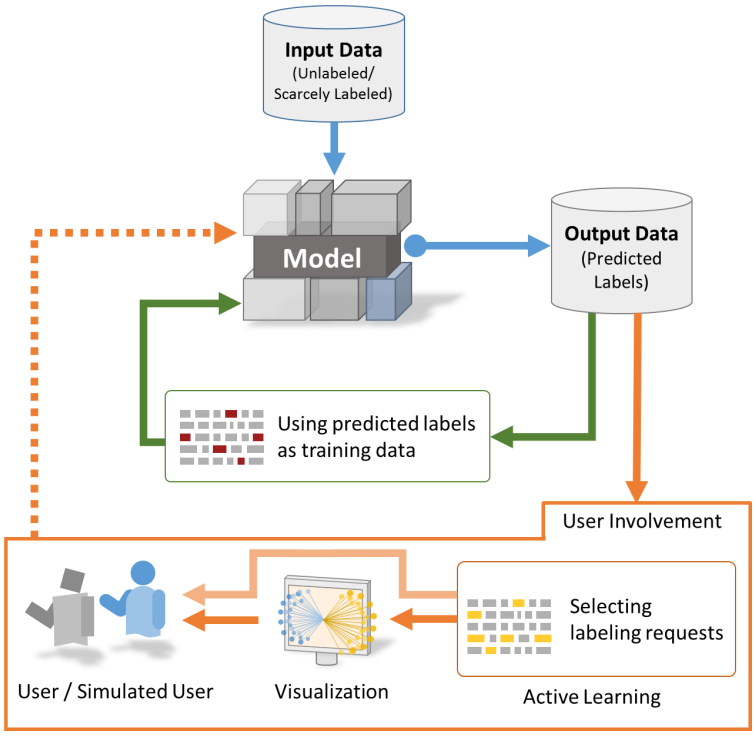


Figure 1.1: Proposed generic framework for interactive text analytics of user-generated data

Central to this thesis is the proposal of a general framework for these objectives, which is illustrated in Fig. 1.1. The model in this framework presents the solution for a particular task, which can be a retrieval, classification, clustering, or ranking model. On one hand, automatic techniques, such as pseudo-relevance feedback, can be used to improve the performance of the model (shown with green arrows). In this case, predicted instances that are the most probable to be correct are used to train the model. However, this approach alone exhibits the limitations of automatic methods, i.e. updating the model based on wrong labels or assumptions of the instances. On the other hand, if a user, who provides feedback for selected instances, is available, the model can be updated based on the provided labels (shown with orange arrows). As mentioned earlier, the idea of employing visualization and active learning techniques is to make user involvement more efficient.

Our objectives in this thesis include:

- Improving the performance of processing user-generated content to an extent that cannot be reached with automatic methods by incorporating user knowledge in the analytical process.
- Proposing novel active learning strategies and interactive visualizations that are based on the characteristics of the data and the task, aiming at gaining the most from the user engagement while minimizing the supervision effort, and augmenting user's analytical skills for their sense-making of the data.
- Evaluating the effectiveness of the proposed techniques for improving the performance of the text mining methods and in comparison with state-of-the-art methods.

We aim to meet these objectives through addressing the following tasks: 1) sentiment lexicon construction for Web opinion data, 2) microblog retrieval for real-time events, and 3) microblog filtering based on user interest profiles.

1.4 Outline

Online customer reviews are an important type of user-generated content for customers and product managers as they contain consumers' viewpoints and their satisfactions or criticisms about different products and services. Potential customers use reviews for making a better purchase and managers analyze this data for building effective customer strategies. Text analytics tools that assist users in these tasks require high quality sentiment lexicons. Since sentiment words have different polarities not only in different domains, but also in different contexts within the same domain, constructing such lexicons is not an easy task. The problem of constructing context-specific sentiment lexicons for online opinion data is discussed in Chapter 2. We apply an automatic method to extract the sentiment pairs from the text and predict their polarities. Then, those pairs with the least certainty values are selected to be labeled by the user. We propose an interactive visual interface and a number of strategies to facilitate user involvement in the labeling process. The visual interface is evaluated in a user study. Chapter 2 is an extended version of our research article [176], which is augmented with the user study and statistical analysis of its results.

Another type of online user-generated content are social networking posts such as

tweets. Tweets are online short text messages posted by Twitter [271] users, which can be rich sources of information for recent affairs. In Chapter 3, we discuss the association of online short text messages with political debates. Short text categorization is a challenging task that has recently become of significance considering the rapid growth of concise online communications. Extracting meaningful information from this sheer volume of data has many applications. We focus on the problem of topical classification of tweets and its challenges. In addition, we propose novel active learning strategies based on the specific features of the data, which improves the quality of the results. A visual interface is developed, which enables users to explore the data and provide their feedback using available interactions. The visual interface is evaluated by three domain experts and its functionality is showcased by several use cases considering different datasets. Chapter 3 presents the main contributions from two research articles [178, 177].

With the vast number and variety of discussed topics on social networking sites, users of these platforms struggle to find relevant information to their interests. Chapter 4 addresses the task of microblog information filtering based on user interest profiles. We start with formulating queries from the user profiles and employing an information retrieval method for retrieving relevant tweets. To tackle the problem of vocabulary mismatch between relevant microblog posts and the queries, we propose automatic and active query expansion techniques, using semantic relatedness methods. The results demonstrate that the proposed active query expansion strategies, which involve the user in the process, improve the performance of our filtering system significantly, while outperforming the state-of-the-art methods. Chapter 4 is a significantly extended version of our research articles [179, 175].

A summary of this thesis is discussed in Chapter 5 which reviews our findings and contributions in each text mining task. We also briefly present our future research directions, which extend beyond the objectives of this thesis.

Chapter 2

Context-Specific Sentiment Lexicon Construction for Sentiment Analysis

2.1 Introduction

With the growth of Web opinion data, the need for analyzing people's attitudes toward different topics has increased markedly. For instance, companies use reviews for handling customer service issues, engaging with customers and finding novel ideas in order to promote their products. In most existing automatic sentiment analysis methods, utilizing a comprehensive sentiment lexicon is crucial; otherwise the intended sentiment could be misinterpreted. However, we know that the sentiment associated to the words is dependent not only on the topic domain but also on the context. For instance, in the domain of cell phones, "high" has a negative sense for the "price" aspect while it has a positive sense for the "quality" aspect. Therefore, we can state that even in a same domain, the same word may have different polarities for different aspects. That is to say, the polarity of a sentiment word is often context-dependent [61].

Therefore, available general-purpose sentiment lexicons cannot be optimal for domain dependent sentiment analysis applications, as these lexicons cannot cover sentiment words for all different domains [170]. In addition, considering the usage context of the sentiment words when determining their polarity can improve the sentiment detection [286]. Therefore, utilizing context-specific sentiment lexicons can improve the accuracy of opinion mining applications. Since manually constructing these lexicons is a hard, tedious and time-consuming task, researchers have studied automatic methods for creating domain and context-dependent sentiment lexicons [170]. However, these methods may encounter ambiguous cases with contradictory evidences, where it is difficult to automatically predict the polarity of the sentiment words. This motivates us to involve the user in the process of constructing such lexicons in order

to benefit from her knowledge to further improve the accuracy of automatic methods.

2.1.1 Research Problem

The objective of the methods proposed in this chapter is to generate context-specific sentiment lexicons for different products from their online reviews. Entries in context-specific sentiment lexicons are pairs of sentiment words and aspects. Sentiment words indicate sentiment polarities such as positive or negative, while aspects are noun phrases that are modified by the sentiment words. For example, one entry can be "(huge, price)". In this thesis, we refer to these pairs as sentiment pairs. We define our research problem as follows:

Definition 1 *Given a set of customer reviews* $D = \{d_1, d_2, ..., d_m\}$ *about a particular product, the task is to create a sentiment lexicon* $L = \{(s_i, a_j, p_{i,j})\}$*, such that* s_i *is a sentiment word modifying aspect* a_j *of the product, and* $p_{i,j}$ *is the polarity of* s_i *with regard to* a_j *indicating how positive or negative the sentiment pair* (s_i, a_j) *is, which means that* L *can contain both* $(s_i, a_j, p_{i,j})$ *and* $(s_j, a_{j'}, p_{i,j'})$*, where if* $j \neq j'$ *then* $p_{i,j}$ *can be different than* $p_{i,j'}$*.*

We intend to automatically generate these lexicons and improve their quality by incorporating an information source in the process of the polarity assignment. The information source can provide the true polarity of a sentiment pair at a defined cost. Therefore, we aim at designing a customized visual interface to reduce supervision effort and make the polarity assignment task easier for human users. In addition, a user study is required to evaluate to what extent involving the user in the process of polarity assignment improves the quality of the lexicon and whether the visual interface is helpful. An overview of the proposed approach is illustrated in Fig. 2.1, whose steps will be explained throughout this chapter.

Briefly, the main contributions of this chapter are:

- Improving the quality of context-specific sentiment lexicons by engaging the user in the polarity assignment process and determining the extent to which this improvement is possible.
- Introducing a novel visualization for constructing context-dependent sentiment lexicons with the following capabilities: 1) presenting the extracted sentiment

pairs and their polarities predicted by the automatic algorithm, 2) providing interactions that enable the user to assign new polarities to sentiment pairs, and 3) making user involvement easier by categorizing aspects and presenting sentiment pairs in a structured way.

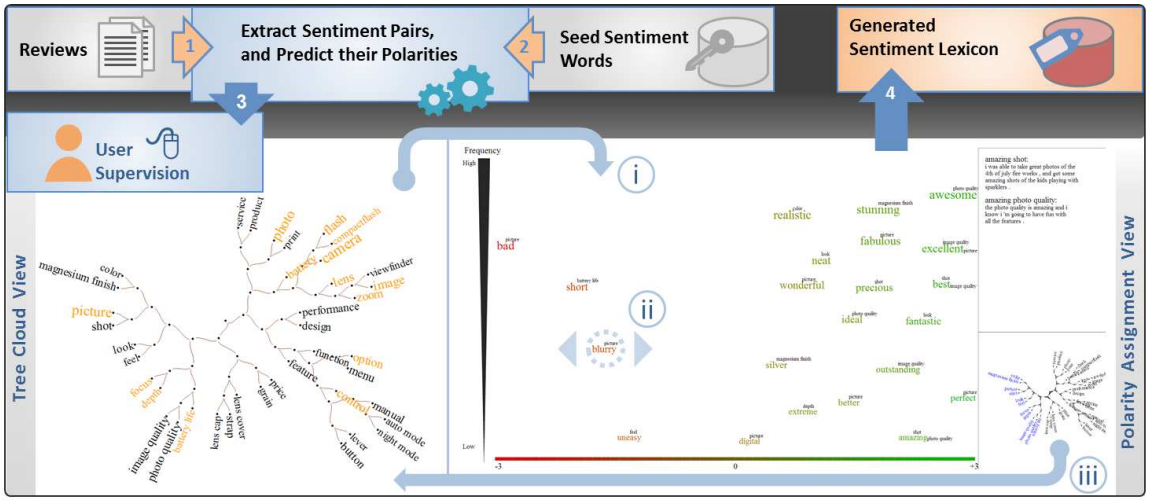


Figure 2.1: An overview of the proposed approach with the visual interface. The automatic algorithm generates the sentiment lexicon and the user improves its quality through the visual interface. Labels i and iii show navigation between two views of the visual interface, and ii show a sentiment pair in the polarity assignment view.

2.1.2 Overview

This chapter addresses the problem of constructing context-specific sentiment lexicons by employing a semi-supervised method for automatically generating sentiment lexicons and involving the user in order to improve the quality of the generated results. Following our proposed framework in Chapter 1, we employ visualization techniques to facilitate user involvement and demonstrate in a user study that the visual interface makes the supervision task easier compared to a text-based interface. This chapter is organized as follows. A survey of related work is provided in Section 2.2. The automatic method for generating sentiment lexicons and the proposed visual interface are described in Section 2.3. The user study and the analysis of its results are discussed in Section 2.4. Finally, Section 2.5 contains the conclusions on our study of supporting the generation of context-specific sentiment lexicons and future work.

2.2 Related Work

As we use visualization for constructing sentiment lexicons, we review the literature in two subsections: first, work related to sentiment lexicon extraction and then research related to visualizing sentiment values. In addition, since creating sentiment lexicons and sentiment analysis are two related problems with similar solutions, we briefly discuss recent research on sentiment analysis at the end of this section.

2.2.1 Generating Sentiment Lexicons

As explained previously in the introduction section, the need for creating domain adapted sentiment lexicons has led researchers toward the proposal of automatic methods for constructing these lexicons. Most of the existing methods use "seed" words, which have known polarity, to calculate the sentiment value of the unknown words. These approaches typically use a general purpose sentiment lexicon such as MPQA [196] or HGI [108] along with methods to propagate the sentiment of the known seed words to the unknown sentiment words. Different propagation approaches have been used. Some methods are based on the context coherence and linguistic heuristics in the context. For instance, if two sentiment words are linked by the conjunction "and", they tend to have the same sentiment polarity, but if they are linked by the preposition "but", they most likely have opposite polarities [136]. Some other methods are based on the co-occurrence frequency of the unknown sentiment word and the seed words. For example, Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA) are used to compute how an unknown word is correlated with seed words like "excellent" or "bad" [270]. In addition, novel statistical models for extracting and clustering aspects from a corpus using some seed words was proposed [198].

Several methods use a dependency grammar to exploit the relationships between sentiment words and aspects. Three types of relationships were used in the double propagation method proposed by Qui et al. [225]. These are: the relationship between sentiment words and aspects, between aspects themselves, and between sentiment words themselves. Using dependency trees that show these relations between words makes it feasible to extract new sentiment words. To compute the polarity of

the newly extracted sentiment words, sentiment values are propagated through both sentiment words and the features [225, 226]. In addition, there are methods that utilize a parser and a large background corpus to extract syntactic contexts of clue words. Top syntactic contexts with the highest entropy were selected and used to discover potential aspects. For instance, the chi-square metric was used to select top aspects and discover sentiment words that co-occur with these aspects [130]. However, this method does not compute the polarities of the discovered sentiment words.

A different strategy is to use existing general-purpose sentiment lexicons along with a method that can adapt the lexicon to the new domain [45]. There is a study that applies a cross-domain classifier to construct domain-dependent sentiment and aspect lexicons with no training data [156]. That work employs a relational adaptive bootstrapping method to propagate information from a source domain with lots of labeled data to a target domain with no labeled data. This information includes the labeled data in the source domain and the relationships between sentiment and aspect words. The results show that the method is comparable with supervised methods, but it is worth mentioning that it only extracts the sentiment pairs and does not assign any polarity to them. Another recent work used word embeddings aligned across languages to translate sentiment lexicons from a source language to a target language [240]. A bilingual word graph was proposed by Gao et al. [84] for the task of cross-lingual sentiment lexicon learning, which leverages intra-language relations among the words in the same language and inter-language relations among the words between different languages in generating a sentiment lexicon for the target language.

In addition to corpus-based methods that use co-occurrence statistics, there are some methods that make use of knowledge sources like WordNet [223] to expand the sentiment lexicon for different domains. These methods typically use distance to the seed words, synonyms and antonyms to calculate the polarity of adjectives [135, 230]. The gloss information available in dictionaries can also be used in polarity assignment. For instance, assuming semantic orientations (positive and negative) as spins of electrons (up and down), the mean field approximation was employed for automatic creation of sentiment lexicons from glosses in a dictionary [260]. Deng et al. [56] used an unlabeled corpus and a dictionary to adapt existing sentiment lexicons for domain-specific sentiment analysis of social media. In addition, there is

SentiWordNet [67], which uses glosses associated to synsets to assign three numerical scores to each synset in WordNet. Numerical scores assigned to each synset indicate how objective, positive, and negative the terms in a synset are [67]. Additionally, there are methods that use Wikipedia [288] to extract aspects in a domain and then employ a bootstrapping method to assign the polarity to the aspect-dependent sentiment words [69].

Often, product reviews have some meta-data. There are some approaches that use this meta-data to construct the sentiment lexicon. For example, the list of pros and cons was used to form a labeled dataset and train a classifier, which in turn was used to extract pros and cons from reviews without this meta-data [144]. In addition, noisy keyphrase lists of pros and cons can be used to infer the semantic properties of documents [27]. More related to our proposed method, is the recent work by Broß and Ehrig [30]. They used the list of pros and cons of reviews as implicit indicators of positive and negative sentiment in order to augment an existing sentiment lexicon by adding context-aware sentiment pairs to it. Their proposed method is based on the heuristics to relate aspects to sentiment words and uses co-occurrence statistics of sentiment pairs in the list of pros or cons for determining their polarity values.

Finally, there are some approaches that combine more than one of the previously mentioned methods. It can be a combination of linguistic heuristics, intra-sentence and inter-sentence conjunction rules, with antonym and synonym rules [61], or a combination of bootstrapping model and corpus-based strategies [14]. An optimization framework that combines different information sources such as a general-purpose sentiment lexicon, an overall sentiment rating at the document level, a thesaurus such as WordNet, and linguistic heuristics was proposed by Lu et al. [170]. Their method automatically discovers sentiment words that have different polarities with respect to different aspects, and although it has higher precision and recall compared to the methods that consider only the overall ratings of the reviews, there is still room for improvement in discovering sentiment words and calculating their polarities.

The method that we propose in this chapter differs from previous work in several ways. First, we involve the user in the polarity assignment process. To the best of our knowledge, there is no prior formal work that engages the user to improve the results of automatic algorithms that generate context-specific sentiment lexicons.

There is only one study on involving the user in the extraction of the aspects in a corpus of reviews [146], but no user study or evaluation has been reported. Another difference with our approach is that users are not involved in the polarity assignment. Though we note that after our work was published [176], an active algorithm for generating domain-specific sentiment lexicons was proposed [212] by Park et al. They involved the user for assigning sentiment labels at the document level and employed a generative probabilistic model to derive the sentiment labels at the word level from that assignment. The article by Park et al. differs from ours in that no visualization technique has been used for user supervision. Furthermore, that method results in domain-specific sentiment lexicons by assigning polarity to sentiment words, while we engage the user in creating context-specific sentiment lexicons by assigning polarity to sentiment pairs.

2.2.2 Visualizing Sentiment Words

Related works that use visual interfaces for presenting sentiment values are reviewed in this section. Researchers have used visualization in different sentiment analysis applications. However, most of them just presented the results rather than providing an interface with interactive capabilities to enable users to provide input.

Tree Maps have been commonly used to present clusters of topics and their sentiments. For instance, Pulse [82] is an interactive visual interface in which clusters of topics extracted from customer reviews are shown by boxes and the color of each box is determined by the average sentiment values of the sentences that belong to that cluster. In this last work, the color varies from red to green to encode negative and positive sentiment values, respectively. Tree Maps have been also used to show the hierarchical structure of the extracted information, where size and color of the rectangles can be used to encode the importance and the customer opinion of the features, respectively [35].

Using colors for encoding sentiment polarities is fairly common. For instance, green, red, and yellow were used to show positive, negative, and neutral movies respectively after SVM was employed to classify movie blogs based on their sentiment polarities [11]. Besides showing sentiment polarities, there are systems that visually encode uncertainty values [291]. In addition, color has been used for showing stance.

As an example, a visual analytics tool was proposed for stance analysis of online social media text [147]. PEARL is a visual analytics tool for visualizing changes in users' emotions from their tweets [306]. While vertical position and brightness are used for representing the valence and the arousal values respectively, color indicates different types of emotion. A tool for visualizing users' reactions to public events using a fine-grained, multi-category emotion model, where emotional categories are also color-coded is EmotionWatch [141]. As a result of this review, we decided to also employ colors to encode the sentiment polarity. In addition, tree clouds and tag clouds are used to present extracted information about sentiment pairs.

Various visual metaphors have been employed to visualize the sentiment content of documents. In order to show more affects than just positivity and negativity, a metaphor inspired by rose plots was proposed by Gregory et al. [96]. In this visualization, each affect is paired with its opposite in order to allow direct comparisons. In addition, each pair has a unique color and the intensity is used to encode the positivity and negativity of the affects within a pair. The concept of a box plot is applied to each petal to present the median and quartile values. Unit circles are also used to present the expected values and the deviation from it (see Fig. 2.2). Bar graphs have been used as another metaphor suitable for comparing reviews. For instance, bar charts were employed in Opinion Observer [161], a visualization tool, to enable users to compare different features of products very easily with a single glance. Moreover, conflicts of opinions have also been visualized by different techniques [39]. A visualization of sentiment conflicts among different social groups was proposed in SocialHelix [34], a visualization that enables users to better understand and analyze when, why, and how these conflicts evolved. SocailHelix is based on a metaphor of a DNA molecule, where two twisting helices are used to visualize conflicting sides (see Fig. 2.3).

There are a number of studies that visualize and track sentiment changes over time. For example, pixel map calendar [105] is employed to present the general overview of a dataset. Each pixel presents a document and the color of the pixel indicates the sentiment polarity of that document. There is also a novel visualization called time density plots, which is based on the occurrence frequency of features and enables users to detect interesting time patterns [236]. In addition, review streams in

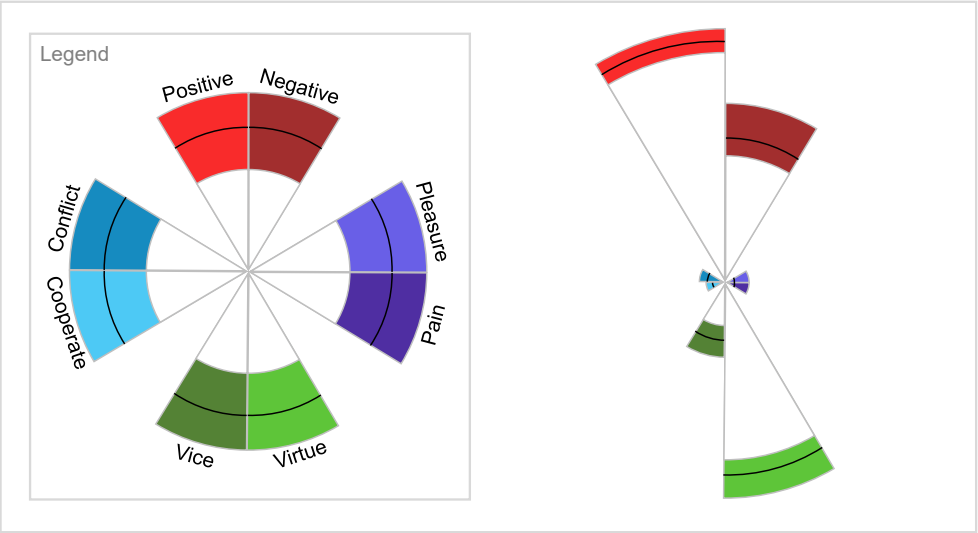


Figure 2.2: Rose plot showing different affects and a petal presenting the range of affect (median and quartile variation) [96]

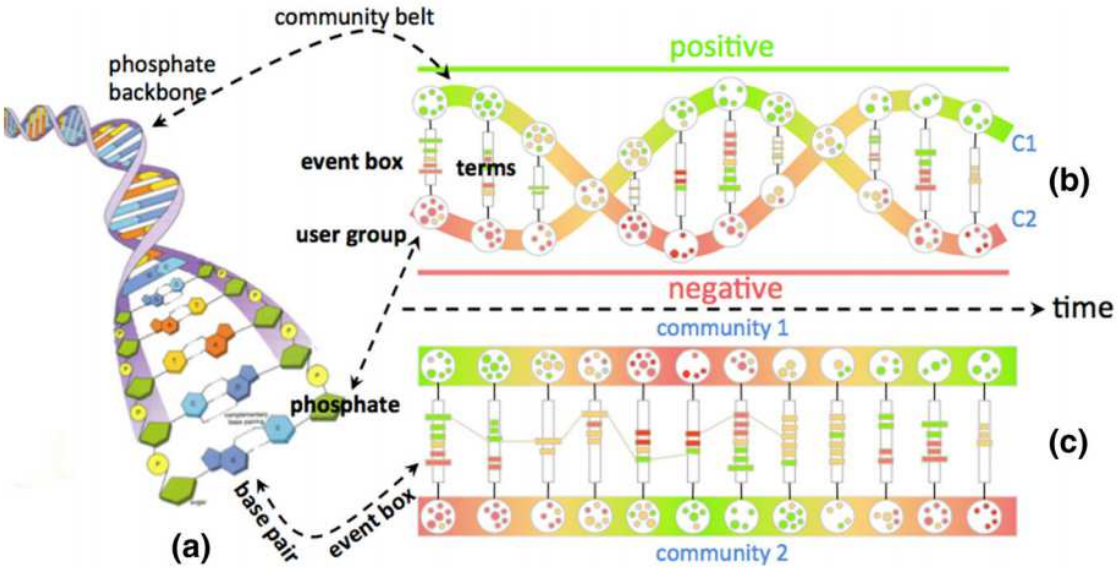


Figure 2.3: SocialHelix: a) mimicking the structure of DNA molecule, b) showing temporal transition of different sentiments, c) the event view presenting the conversation throughout the event [34]

tweets have been explored by combining a new pixel cell-based sentiment calendar, a geo-temporal map and self-organizing maps into an integrated analysis system [106]. Standard line-plots have also been used for showing the changes of stance over time for a particular target in a visual stance analysis tool [147].

line graphs and stream graphs have been used for representing temporal trends and changes of sentiment over time. For instance, they were used in a visual analysis system, Agave [29], that enables users to explore events and analyze their sentiment from Twitter. Helix structure also has been used for illustrating sentiment evolution over time in an interactive system for analyzing sentiments of popular topics [279]. Another interactive visualization for illustrating the opinion propagation among Twitter users and the attention transition of users between different topics is OpinionFlow [290]. OpinionFlow uses Sankey diagrams for visualizing the flow of users across different topics, while density maps present the diffusion of opinions among users. Line charts are also used to present the number of positive and negative reviews over a period of time [186]. In the method proposed in this chapter, we focus on visualizing sentiment words and their polarity for different aspects of a product. We do not consider time in our visual interface assuming that the sentiment of affective words do not change over time so rapidly.

More recently, an interactive visualization for sentiment lexicons was proposed by Chen et al. [44], which in addition to presenting the sentiment pairs, enables users to make corrections to their estimated polarities using drag and drop interactions. Their visual interface presents topical clusters of hotel reviews, which are created using a Latent Dirichlet Allocation (LDA) model, along with extracted sentiment pairs from reviews. In addition to the different techniques used for creating the lexicon and clustering the sentiment pairs, our visual interface presents the clusters of aspects in a separate view and employs strategies for reducing user effort (see Section 2.3.2).

2.2.3 Sentiment Analysis

There are three levels of sentiment analysis: document level, sentence level, and aspect level [162]. The document level considers the whole document (product review) as the processing unit and determines the sentiment polarity for the document as a whole. While the sentence level performs the analysis on individual sentences, the aspect

level determines the sentiment orientation about each aspect or entity of the target item (product).

There are many studies on sentiment analysis of product reviews, from categorizing the polarity of sentiments [70] to ranking the products based on their sentiments [143]. It has been argued that ranking the products based on specific needs is more helpful than summarizing the reviews when users want to select a product [143].

Moreover, automatic detection of the most positive and negative aspects of a product from customer reviews was proposed by Bancken et al. [15]. In that study, a set of pre-defined syntactic dependencies between words was used for extracting opinions about different aspects of a product. Then, a WordNet-based similarity measure was employed to cluster different mentions of the same aspect and calculate an overall sentiment score for each aspect. Our work differs from that study in several ways. In addition to the differences in the automatic method for extracting sentiment words and aspects, we involve the user in the process and propose a visual interface, where useful visual information can be encoded. More detailed summaries of research on sentiment analysis are discussed in several surveys [211, 162, 229, 231, 249, 259].

The growth and popularity of social media platforms have made them important information sources for a vast variety of topics. Performing sentiment analysis on the large amount of data produced by millions of users of these platforms is one of the important success factors of prominent companies [4]. Twitter is one of the largest and most popular microblogging platforms. Some recent studies on sentiment analysis of Twitter are [242, 90, 200, 58, 183, 37].

In addition, there are two ways for representing emotions: categorical and dimensional [32]. Categorical sentiment analysis represents emotions as a set of discrete categories, while in the dimensional sentiment analysis, emotions are represented as a low dimensional continuous space. The most common model is the valence-arousal space, where valence describes pleasure/displeasure or positive/negative, and arousal indicates the activation/deactivation level. Based on this model, emotions can be represented as points in the valence-arousal coordinate plane [300]. In recent works, approaches for determining the valence-arousal values of sentiment words of one language from those of another language have been proposed [280, 296]. In addition, a Chinese valence-arousal lexicon was created manually by five annotators [299].

Table 2.1: Notation used in this chapter

Notation	Description
D	Set of customer reviews for a particular produc.
S	Set of sentiment words
$\widetilde{S}$	Set of seed sentiment words
s_i	A sentiment word, which can be a seed sentiment word or a context-dependent sentiment word
A	Set of aspects
a_j	An aspect of the product
C	Set of extracted sentiment pairs from reviews
(s_i, a_j)	A sentiment pair with sentiment word s_i and aspect a_j
$p_{i,j}$	Polarity of sentiment word s_i when modifying aspect a_j
L	Sentiment lexicon, which assigns a polarity $p_{i,j}$ to each sentiment pair $(s_i, a_j) \in C$
R	Set of relation types for extracting sentiment pairs
E	Extracted dependency relations from reviews
N	List of part-of-speech tags for aspects
J	List of part-of-speech tags for sentiment words
(g_k, d_k, e_k)	A triplet showing a dependency relation between governor g_k and dependent d_k, when e_k is the relation type

Finally, comparative opinion mining and stance detection are two related problems, that have recently received a lot of attention, yet are different from sentiment analysis. Comparative opinion mining focuses on extracting comparative relations between entities from opinions [83, 293, 253, 275], while stance detection and classification is about determining the stance of the author of the text with respect to a target, i.e. whether the author is "in favor", "against", or "neutral" about the target [193, 65, 194].

2.3 Proposed Method

We describe our proposed method in two sections: 1) the automatic method for extracting sentiment pairs and assigning polarity values to them, and 2) the visual interface, which the user interacts with. The notation used in this chapter is summarized in Table 2.1.

2.3.1 Automatic Lexicon Creation

First, we explain how the sentiment pairs are discovered, and then we discuss the calculation of their polarity.

Extracting Sentiment Pairs

The method for extracting sentiment pairs is based on this fact that sentiment words and their targets are linked by several syntactic relations [226], which can be explored by a dependency parser. Consequently, having an initial list of seed sentiment words, one can attempt to discover existing sentiment pairs in an iterative process. This is inspired by the work of Qui et al. [226], where dependency relations between sentiment words and aspects are used in an iterative process to extract sentiment pairs from the reviews. In each iteration, the lists of known sentiment words and aspects are expanded as new sentiment pairs are found. These newly updated lists are used to extract more sentiment pairs in the same way. This strategy is consistent with our proposed generic framework in Chapter 1, where predicted data by the automatic method is used for updating the method itself. It is important to note that the proposed method by Qui et al. constructs a domain-specific sentiment lexicon rather than a context-specific sentiment lexicon, which means that sentiment words have the same polarity for all different aspects within a domain. Therefore, we adapted the proposed method to make it suitable for constructing context-dependent sentiment lexicons.

We considered nouns and noun phrases to be aspects, while sentiment words are terms modifying these aspects. To extract sentiment pairs from reviews, we utilize the Stanford parser [54, 98] to extract the dependency relations between different words in the reviews. Stanford dependencies define a binary relation between two words and consist of type of the relation, governor and dependent, where governor is usually the head of the dependency relation and dependent is the modifier of the relation. Extracted relations from reviews are matched with a set of rules, which describe the dependency relations between sentiment words and aspects. The list of dependency rules, which was extracted from Stanford typed dependencies manual [55], is shown in Table 2.2. For instance, “amod” or “adjectival modifier” indicates an adjective

modifying a noun phrase. When an extracted relation from reviews and the part-of-speech tags of the words in that dependency relation matches with one of these dependencies rules, a sentiment pair is constructed and added to the lexicon. These steps are shown in Fig. 2.4.

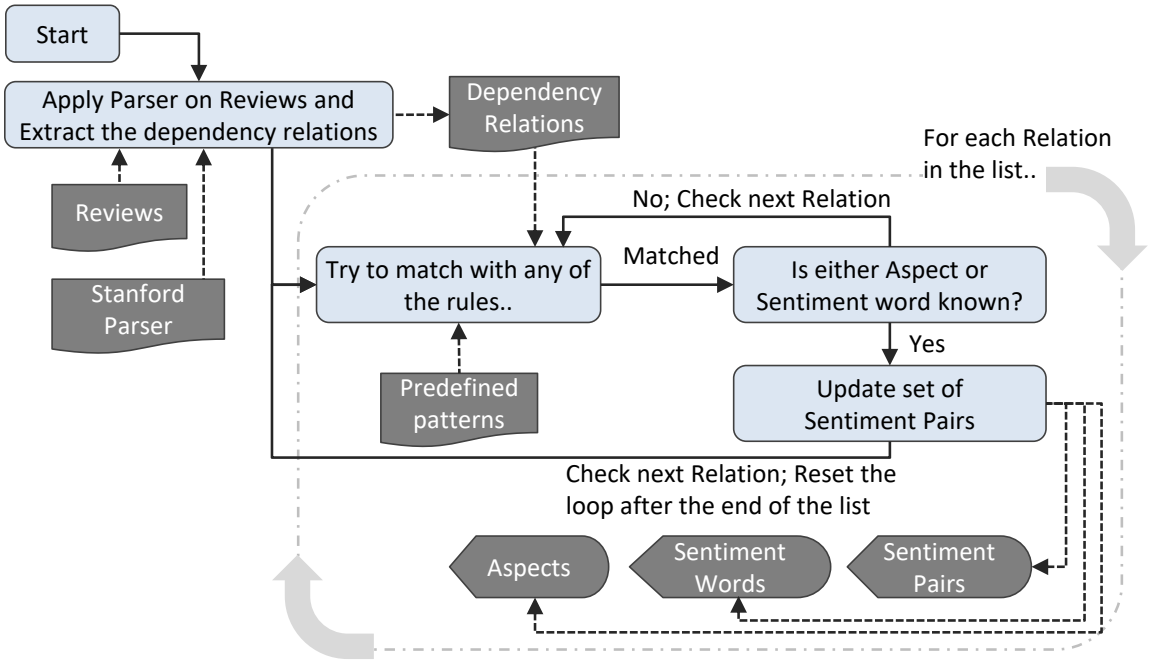


Figure 2.4: Extracting sentiment pairs from the set of reviews

Specifically, we follow Algorithm 1 to extract sentiment pairs. In these steps, $A = \{a_j\}$ and $S = \{s_i\}$ indicate the list of aspects and sentiment words, respectively. $C = \{(s_i, a_j)\}$ denotes the extracted sentiment pairs, where s_i is a sentiment word and a_j is an aspect. $N = \{NNS, NNP, NN, NNPS\}$ and $J = \{JJ, JJS, JJR\}$ are the acceptable part-of-speech tags for aspects and sentiment words respectively, where NN = noun (singular or mass), NNS = plural noun, NNP = singular proper noun, NNPS = plural proper noun, and JJ = adjective, JJS = superlative adjective, JJR = comparative adjective. E is the set of extracted dependency relations from the reviews. Each triple (g_k, d_k, e_k) shows a dependency relation, where e_k is the type of the dependency relation, and its governor and dependent are shown as g_k and d_k in relation (g_k, d_k, e_k), respectively. In the relations in Table 2.2, g_k is the aspect and d_k is the sentiment word, except for "nsubj" relation type.

Table 2.2: Dependency relations used for extracting sentiment pairs

Name	Description	Example
amod	an adjectival modifier of an noun phrase	removable and cheap battery (removable⟶amod⟶battery)
nsubj	a nominal subject is a noun phrase which is the syntactic subject of a clause	the scroll wheel is finicky (wheel⟶nsubj⟶finicky)
rcmod	a relative clause modifying the noun phrase	the software itself, which should be user-friendly (user-friendly⟶rcmod⟶software)

Algorithm 1 Sentiment Pair Extraction

Input: set of acceptable relation types $R = \{r_1, r_2, ..., r_l\}$, list of seed sentiment words $\widetilde{S}$, set of reviews $D = \{d_1, d_2, ..., d_m\}$

Output: List of extracted sentiment pairs $C = \{(s_i, a_j)\}$.

$E = \{(g_1, d_1, e_1), ..., (g_n, d_n, e_n)\}$ ▷ extracted dependency relations from reviews

$A = \emptyset$, $C = \emptyset$ ▷ A = set of aspects, C = set of sentiment pairs

$S = \widetilde{S}$ ▷ S = set of sentiment words, which is initialized with $\widetilde{S}$

$terminate = false$ ▷ a variable for stopping the algorithm

while $NOT(terminate)$ **do**

  $terminate = true$

  for $(g_k, d_k, e_k) \in E$ **do**

    if $(e_k \in R) \wedge (d_k \in S) \wedge (POS(d_k) \in J) \wedge (POS(g_k) \in N) \wedge (g_k \notin A)$ **then**

      ▷ a new aspect g_k is found based on the already added sentiment word s_k

      $A = A \cup \{g_k\}$, $C = C \cup \{(d_k, g_k)\}$

      $terminate = false$

    if $(e_k \in R) \wedge (g_k \in A) \wedge (POS(g_k) \in N) \wedge (POS(d_k) \in J) \wedge (d_k \notin S)$ **then**

      $S = S \cup \{d_k\}$, $C = C \cup \{(d_k, g_k)\}$

      ▷ a new sentiment word d_k is found based on the already added aspect g_k

      $terminate = false$

    if $(e_k == nsubj) \wedge (POS(d_k) \in N) \wedge (POS(g_k) \in J)$ **then**

      if $(g_k \in S) \wedge (d_k \notin A)$ **then** ▷ relation type is "nsubj" and g_k is already in S

        $A = A \cup \{d_k\}$, $C = C \cup \{(g_k, d_k)\}$ ▷ add d_k as a newly found aspect

        $terminate = false$

      if $(d_k \in A) \wedge (g_k \notin S)$ **then** ▷ relation type is "nsubj" and d_k is already in A

        $S = S \cup \{g_k\}$, $C = C \cup \{(g_k, d_k)\}$ ▷ add g_k as a newly found sentiment word

        $terminate = false$

Polarity Assignment and Selecting Uncertain Pairs

In this thesis, polarity assignment is done at the same time as sentiment pairs are extracted. After a new sentiment pair is discovered, we predict its polarity based on the evidence observed in the context which can be any other sentiment word that modifies the same aspect in the same review. It is reasonable to assume that a reviewer does not change her opinion about a specific aspect within a review. Therefore, other sentiment words that modify the same aspect in the review are considered as evidence. For instance, assume the dataset contains reviews about a printer. If we want to predict the polarity of the newly discovered pair "(tiny, buttons)", we look into its context. If we find other pairs such as "(terrible,buttons)" with a known polarity, we take it into account in calculating the polarity of the pair "(tiny,buttons)". Polarity of sentiment pairs can take a value in the range of (-3,+3). Similarly, polarity +3 and -3 are assigned to the positive and negative seed words, respectively. In addition, we make this safe assumption that seed sentiment words have the same polarity for all the aspects, e.g. "great" and "excellent" are always positive, while "awful" and "terrible" are negative, regardless of the domain or aspect.

After identifying evidences, we calculate the polarity of the newly discovered sentiment pair based on the following equation:

$$p_{i,j} = \frac{\sum_{k=1}^{n} f_{k,j} \times p_{k,j}}{\sum_{k=1}^{n} f_{k,j}} \tag{2.1}$$

where $p_{i,j}$ indicates the polarity of sentiment pair (s_i, a_j), and s_i and a_j are the sentiment word and the aspect, respectively. n is the number of found evidences and $f_{k,j}$ is the frequency of k^{th} evidence, i.e. k^{th} sentiment word that modifies aspect a_j. Respectively, $p_{k,j}$ indicates the polarity of that evidence. It is worth mentioning that in this relation, besides the polarity of the evidences, we also take into account the frequency of their occurrences in the corpus. The reason is that it is expected that if a sentiment word does not appear frequently in the context of an aspect, then its weight in calculating the polarity of other pairs with the same aspect should be low.

If there is no evidence in the context of the new sentiment pair, we consider the polarity of its sentiment word in a general-purpose lexicon as its predicted polarity. In this thesis, we used the lexicon introduced in [112], which is available online [160],

as the general-purpose lexicon. Those pairs that have contradictory evidences or no evidence will be selected as ambiguous cases for the automatic algorithm. These pairs have the lowest confidence value since the algorithm did not find enough evidence for predicting their polarities. Selecting sentiment pairs based on their confidence values can be regarded as an active learning technique. These pairs will be shown along with the most frequent sentiment pairs in the visual interface and the user can provide her input about their polarities through available user interactions.

Our polarity assignment module differs from [226] in two ways. First, Qui et al. assigns polarities to the sentiment words regardless of their contexts or the aspects they modify, while our proposed method assigns polarities to the sentiment pairs. In other words, they assign polarity to each sentiment word or feature individually rather than to the pairs. Second, they calculate the polarity of the new pair by summing up the polarity of all the evidences without considering their frequencies.

The automatic method used in this thesis is just a case study to evaluate our proposed interface. While the proposed visualization can be used with any automatic method that generates context-dependent sentiment lexicons. The details of our visual interface are discussed in the next section.

2.3.2 Visualization

The proposed visual interface consists of two views: the tree cloud view and the polarity assignment view. The tree cloud view is a navigation interface for the polarity assignment view and aims at reducing user effort, while the polarity assignment view presents the sentiment pairs and their polarity and enables the user to assign new polarities. A demo video of the proposed visual interface is available online[1]. In this thesis, all the visualizations were implemented in JavaScript using the d3 library [26].

Tree Cloud View

This view presents the set of existing aspects in the domain. For instance, if our dataset contains reviews about printers, then “price”, “quality”, “ppm”, “customer service”, “shipping”, and “cartridge” are some of the aspects in this domain. These aspects are presented in a tree structure that is constructed based on the semantic

[1] `https://drive.google.com/file/d/0Byrh43zBaFKGNndNUzhkNjNzUzQ/view?usp=sharing`

relatedness between its nodes and is called a Tree Cloud. In other words, aspects that are semantically related will appear close to each other (i.e. neighbors in the tree). For example, in the domain of printers, aspects such as "price" and "cost" will appear as neighbors in the tree as well as "support" and "warranty" (as illustrated in Fig. 2.6).

Tree clouds were first introduced by Gambette and Veronis [81]. They used the co-occurrence frequency of each pair of terms to calculate their relatedness. Since reviews are usually short, sufficient context to confidently compute co-occurrence frequencies may not be available. Therefore, instead of co-occurrence frequency, we employ Google tri-grams from the Google Web 1T data set [28] to calculate the semantic relatedness between the two terms [126]. A word n-gram is a contiguous sequence of n words from a given sequence of text. The frequencies of all the tri-grams that start and end with the pair of terms are added together and then they are normalized using the uni-gram frequency of each of the terms. The result is the similarity score which is in the range $[0, 1]$. Therefore, we can calculate the similarity matrix, which shows the relatedness score for existing pair of terms. Then, to build the tree we use the neighbor-joining [243] algorithm. This algorithm requires a matrix showing the distance between each pair of terms. Therefore, we first convert the similarity matrix to a distance matrix by subtracting the similarity values from value 1, and then follow the steps in Algorithm 2 to build the tree.

In each iteration, the two terms with the lowest distance are joined together and considered as a new single node. Consequently, the dimension of the distance matrix is reduced by one. These steps are continued until all the terms are added to the tree. These steps are also illustrated in Fig. 2.5. The branches of the tree cloud can be viewed as different clusters of the extracted aspects due to the fact that the semantic relatedness values between aspects are used in building the tree. Therefore, the closer the nodes are in the tree, the more semantically related they are. For instance, the terms inside the blue circle in Fig. 2.6 form a cluster related to installing the printer and connecting it to the computer.

The user interactions in this view include selecting a group of aspects and the subsequent navigation to the polarity assignment view. For this purpose, the user can click on a node and all the aspects that are its children will be added to the list

Algorithm 2 Tree Cloud Construction

Input: Matrix $[D]_{n\times n}$ ▷ $d_{i,j}$= distance between terms T_i and T_j

n = number of aspects (terms)

Output: A tree cloud constructed from aspects

while $n > 1$ **do**

  calculate symmetric matrix $[Q]$

$$q_{i,j} = (n-2)d(i,j) - \sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k)$$

  $l, m = \underset{0\le i,j\le n, i\neq j}{\operatorname{argmin}} q_{i,j}$ ▷ find two terms with the lowest distance

  create new node, T_u by joining T_l and T_m

  for $k \neq l, m$ **do**

    $d(u,k) = \frac{1}{2}[d(l,k) + d(m,k) - d(l,m)]$ ▷ update the distance matrix $[D]_{n-1\times n-1}$

  $n = n - 1$

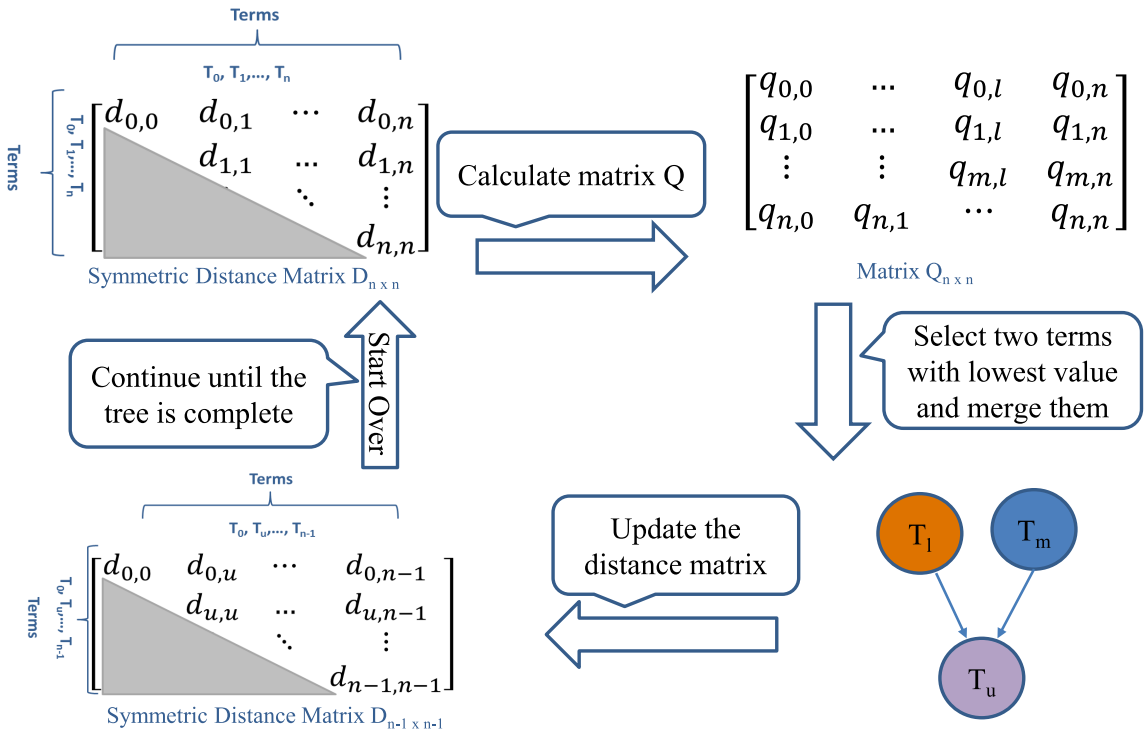


Figure 2.5: An overview of the Neighbor Joining algorithm for constructing the tree from the distance matrix.

of selected aspects. Then, in the polarity assignment view, all the sentiment pairs associated with this set of aspects will be presented. This is illustrated in Fig. 2.7 when sentiment pairs related to the selected aspects, "cost", "price", "print quality", and "quality" are shown. In addition, when the user hovers the mouse over a node, a number appears. This value shows the number of the sentiment pairs that are going to be shown if the user clicks on this node. It can help the user in selecting branches with appropriate size. Some users may prefer more specific topics and select smaller branches, while some may want to see more sentiment pairs together in the next view. Other features of this view such as different color of terms are explained in the next two sections.

Figure 2.6: Tree cloud built from a printer review dataset, yellow terms show aspects with at least one sentiment word with unknown polarity. Branches that are already seen by the user are dimmed grey.

Polarity Assignment View

This view presents sentiment pairs and their predicted polarities by the automatic algorithm. For this purpose, we use color, size, and position of text elements to present extracted information about the sentiment pairs. Sentiment words are shown as main terms and their aspects are presented as a word cloud around them. The position along the x-axis shows the polarity value of the sentiment pairs. In addition, color is also used to encode polarity value. A color spectrum from red to green at the bottom of the view presents the range from the most negative to the most positive value. The position along the y-axis is based on the sentiment word frequency. Frequent terms appear at the top of the view, while sentiment pairs with lower frequency are shown at the bottom. Moreover, since the size of elements is easy for users to interpret, the font-size of the text elements also shows the frequency. Consequently, sentiment words with high frequency will be shown bigger than the low frequent ones. For example, in Fig. 2.7, "poor" is bigger than "high" which indicates that its occurrence frequency in the corpus is higher.

A miniaturized rendering of the tree cloud view is shown at the bottom right of this view. This acts as a mini-map within this drill down view. All the nodes are shown in black except the selected branch, which is in blue. This lets the user know which branch of the tree she is currently interacting with. Whenever the user is satisfied with the polarities, she can click on the minimized tree and go back to the tree cloud view where the current branch is dimmed grey to indicate that it has already been inspected by the user.

To help the user in making decisions about the polarity of the sentiment pairs, their contexts are also displayed in the context view at the top of the minimized tree (Fig. 2.7). Whenever the user hovers the mouse over a sentiment term, a sample sentence randomly selected from its contexts is shown.

We stated that the main point of the polarity assignment view is to enable users to change the polarity of the sentiment pairs. To do this, the user should move the sentiment words horizontally by dragging and dropping them to the desired position. The color of the selected term will accordingly change based on its horizontal position. In addition, as the user moves a sentiment word, its sentiment value will be shown below it. These features are illustrated in Fig. 2.7 and Fig. 2.8. The sentiment

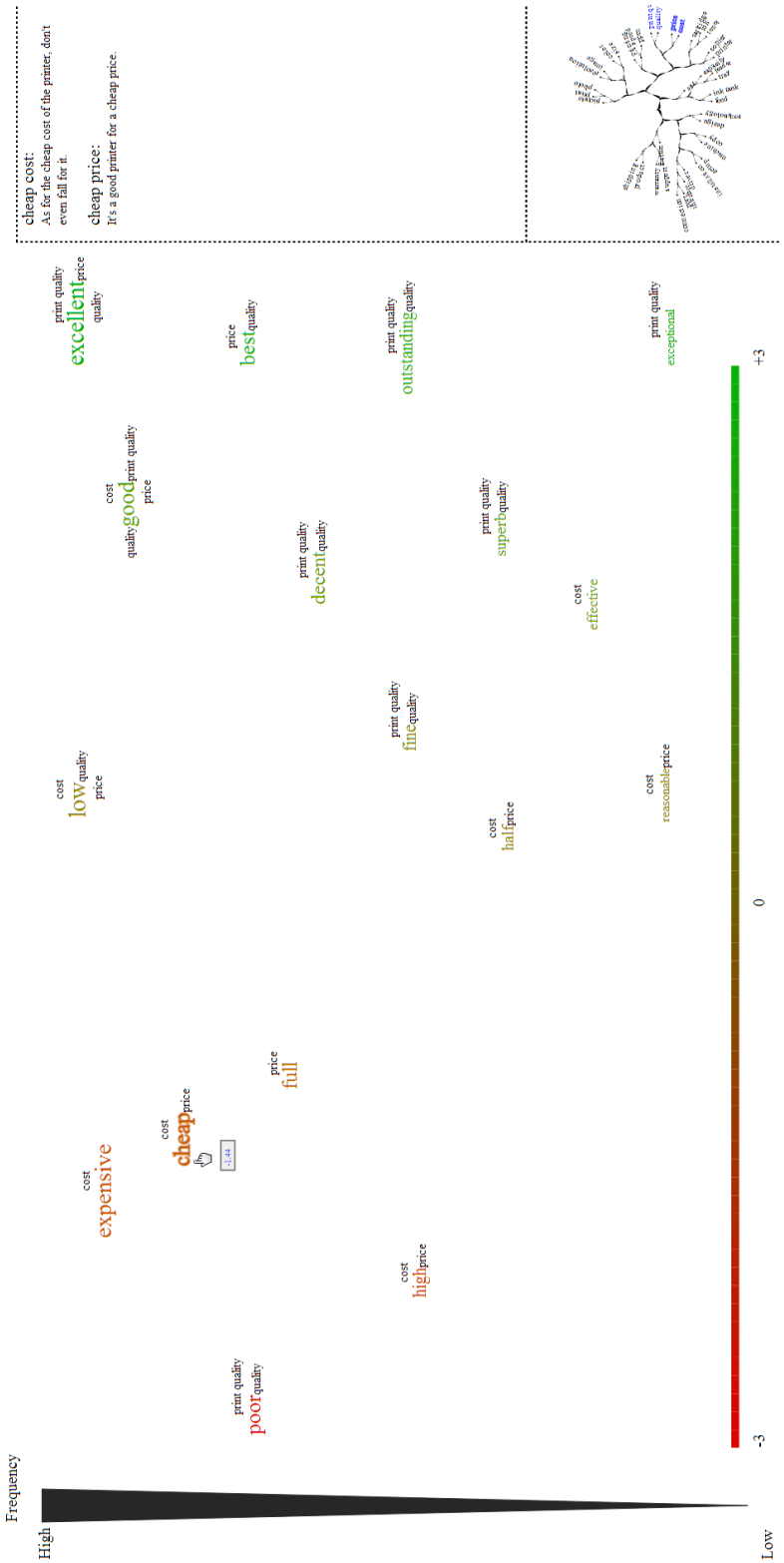


Figure 2.7: Polarity assignment view, sentiment words and aspects are presented as main terms and word clouds, respectively. The color presents the polarity and the size indicates the frequency of the sentiment words. Sample contexts are shown at the top right and the minimized tree is at the bottom right of the screen.

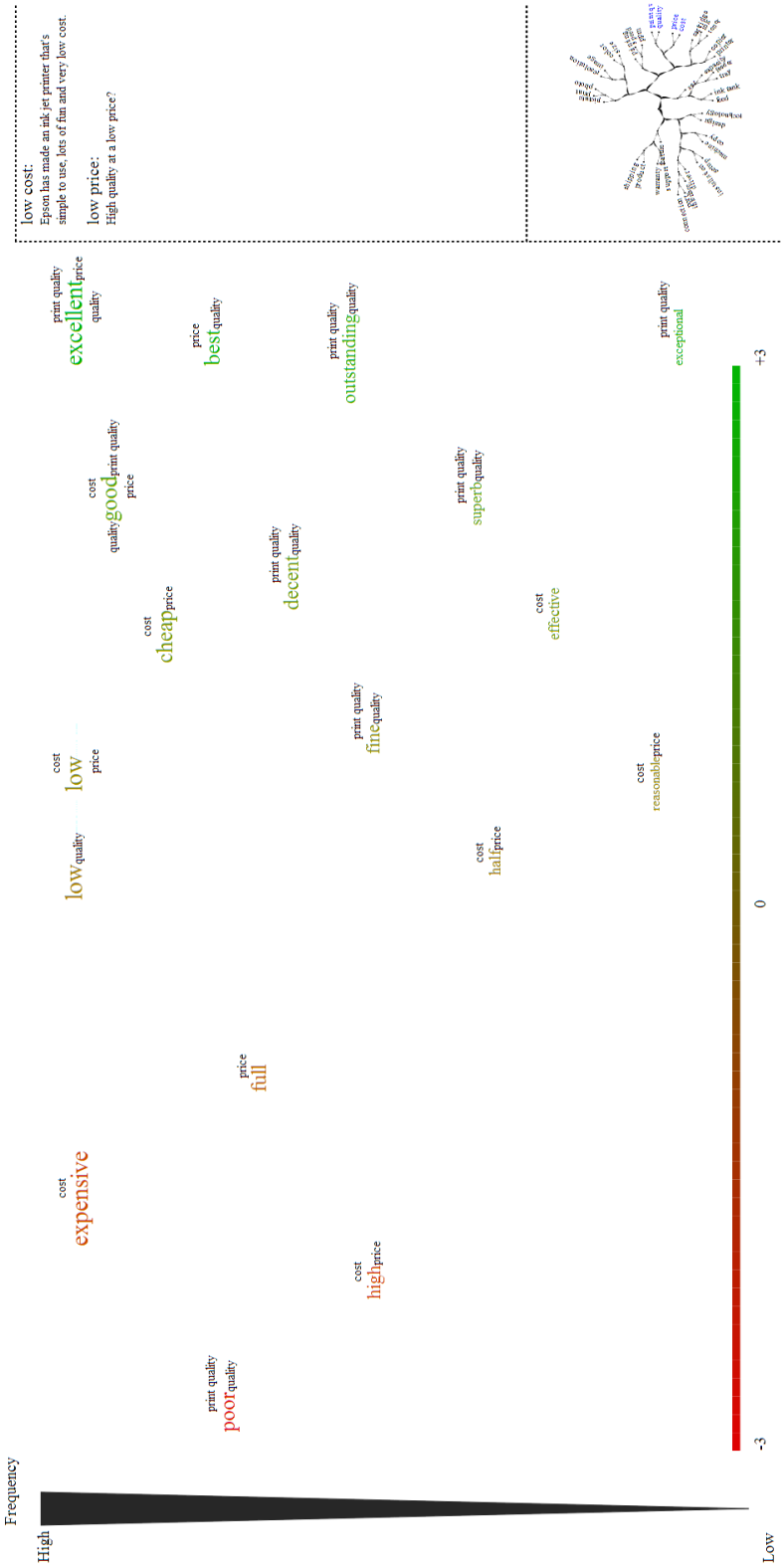


Figure 2.8: Polarity assignment view after the user moved the sentiment word “cheap” and duplicated the sentiment word “low” by clicking on its aspect “quality”.

pairs “(cheap, price)” and “(cheap,cost)” have the same negative sentiment value in Fig. 2.7. After the user moves them with a single interaction, they have positive score in Fig. 2.8. This shows that merging sentiment pairs can reduce the number of interactions.

In the above example, it was shown that a sentiment word may be presented with multiple aspects. That is, multiple sentiment pairs are merged into one node. Another instance is the sentiment word “low”. It is presented with three different aspects, “cost”, “price” and “quality”. The sentiment word “low” appears as positive in this figure. This means that all sentiment pairs “(low, cost)”, “(low, price)” and “(low, quality)” also have positive polarities. Since “low” has a positive sentiment for “cost” and “price” but a negative meaning for “quality”, the user may want to correct the polarity of the pair “(low, quality)”. She can click on the text element presenting “quality” and duplicate the current node. This is illustrated in Fig. 2.9, which presents the duplicated terms along with their associated aspects after the user clicks on “quality”. Now, the user can change the polarity of “(low, quality)” without moving other pairs. The final results of all of these interactions are shown in Fig. 2.9.

As explained earlier, we merge the sentiment pairs to reduce the number of required interactions. However, one may ask why the sentiment pair “(low, quality)” is merged with the other pairs in the above example. Such examples arise because the automatic algorithm predicts its polarity incorrectly. In other words, when two or more sentiment pairs with the same sentiment word have similar polarities, they are merged. Therefore, if the predicted polarity for the sentiment pair “(low, quality)”, were negative, it would have been presented separately.

Reducing User Effort

In this section, we discuss the motivation for this way of visualizing sentiment pairs with regard to reducing user effort. First of all, we believe that presenting aspects organized according to their semantic relatedness makes the task easier. Since the terms appearing as neighbours are semantically close, it is expected that their common sentiment words may have similar polarities. To illustrate, assume that aspects “price” and “cost” are neighbours in the tree. If a sentiment word like “low” has a positive value for one of these aspects, the likelihood that it has the same polarity for

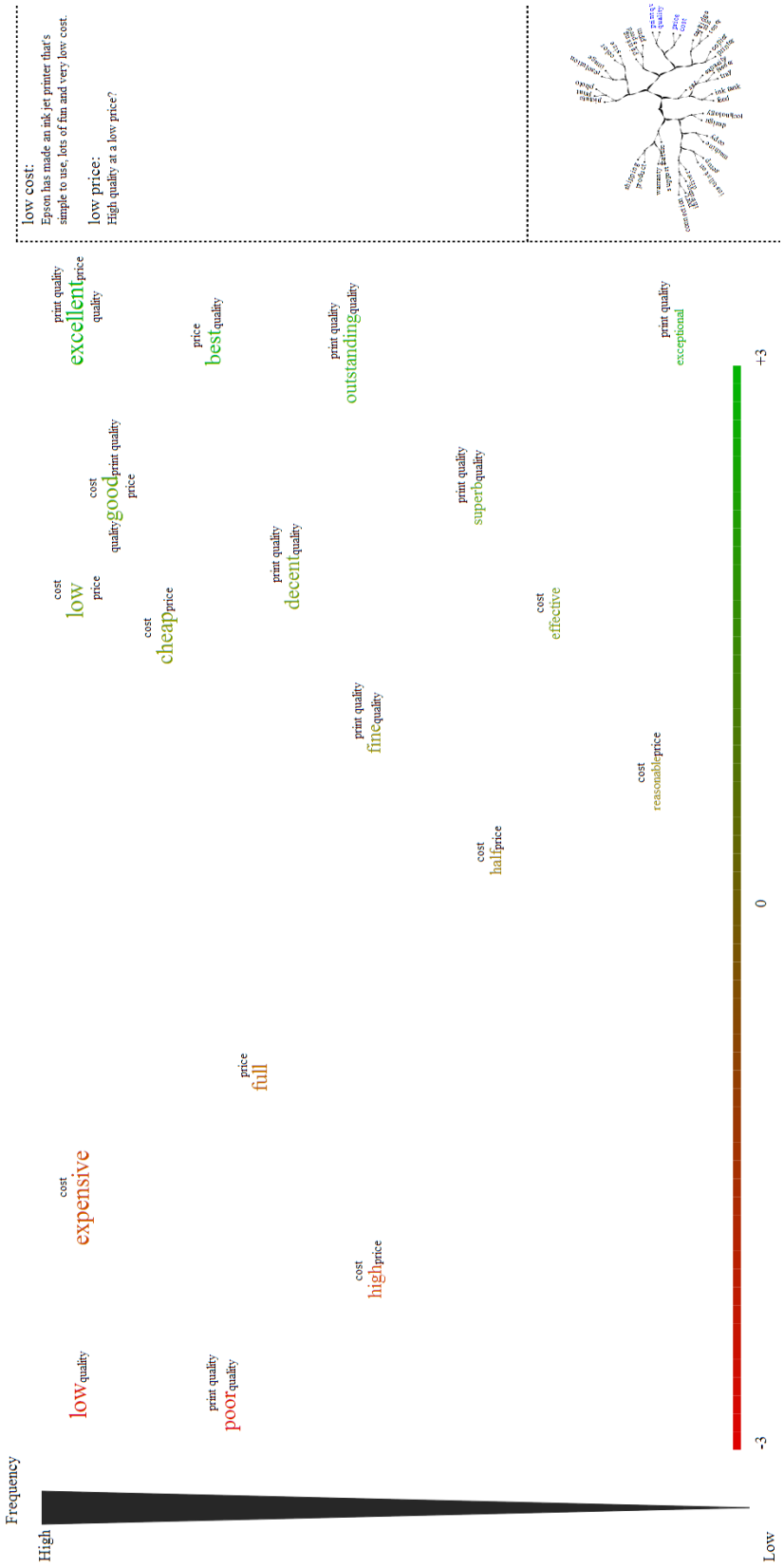


Figure 2.9: Polarity assignment view after the corrections were made by the user

the other one is high. In this case, we can merge these two pairs and present them as a single node, as illustrated in Fig. 2.7. This saves space and also reduces user effort if a polarity change is required. Besides, the tree cloud view provides a means of categorizing sentiment pairs instead of showing all pairs in one view, which would be cluttered and hard to read.

Furthermore, automatic methods generating context-dependent sentiment lexicons may encounter difficult cases in the polarity assignment stage. Therefore, the user input on these sentiment values can improve the quality of the generated lexicon. Aspects that appear in these sentiment pairs will be shown in yellow in the tree cloud, otherwise they will be presented in black. Therefore, the user can spot which aspects contain more ambiguous sentiment pairs, so that she can prioritize them when performing the polarity assignment task. In addition, when the user does not want to view all the sentiment pairs, she can consider only these ambiguous pairs and still improve the quality of the lexicon to a notable extent. Finally, we select the most frequent and ambiguous sentiment pairs and show them to users for supervision instead of showing all the extracted sentiment pairs. This saves users' time when performing the assignment. To evaluate whether these features of the proposed visualization are helpful, we ran a user study and asked the participants to evaluate them. The details of the user study and how it is conducted are discussed in the next section.

2.4 Evaluation

To evaluate the generated lexicon and determine whether the proposed visualization is helpful in the polarity assignment task, we ran a user study. A text-based interface was also implemented to be compared against our visual interface in the user study.

2.4.1 Dataset

Since researchers in this field usually report the performance of their proposed methods on different datasets, mostly gathered and labeled by themselves [298, 161], there is no benchmark dataset for the problem of aspect-based opinion mining [192] and it is difficult to compare the results and find the best method. In this thesis, we used one publicly available dataset whose labels were made also available, so that experiments can be reproducible in the future. The dataset we used for this study

contains reviews about five different products from Amazon. This dataset is available online [160] and have been provided and used by Hu and Liu [112, 113]. The reviews come with labels at the aspect-level, i.e. aspects are labeled with their polarity value. The maximum and minimum values of these labels are +3 and -3, respectively. This dataset is relatively small, each domain contains less than a hundred reviews, and similar in topic, related to electronic devices. The reason for choosing this dataset was to have score labels at the aspect level, so that we can build the gold standard and evaluate our approach. Having more diverse and larger datasets would help improve the evaluation.

2.4.2 Alternative Interface

The text-based interface present the sentiment pairs in a list. This list is ranked based on the frequency of the sentiment pairs. Each row contains one sentiment pair, a slider, and a sample sentence related to the sentiment pair. The value of the slider shows the predicted polarity by the automatic algorithm and it is in the range (-3,+3). The user can change the polarity of the sentiment value by moving the slider. This interface is shown in Fig. 2.10.

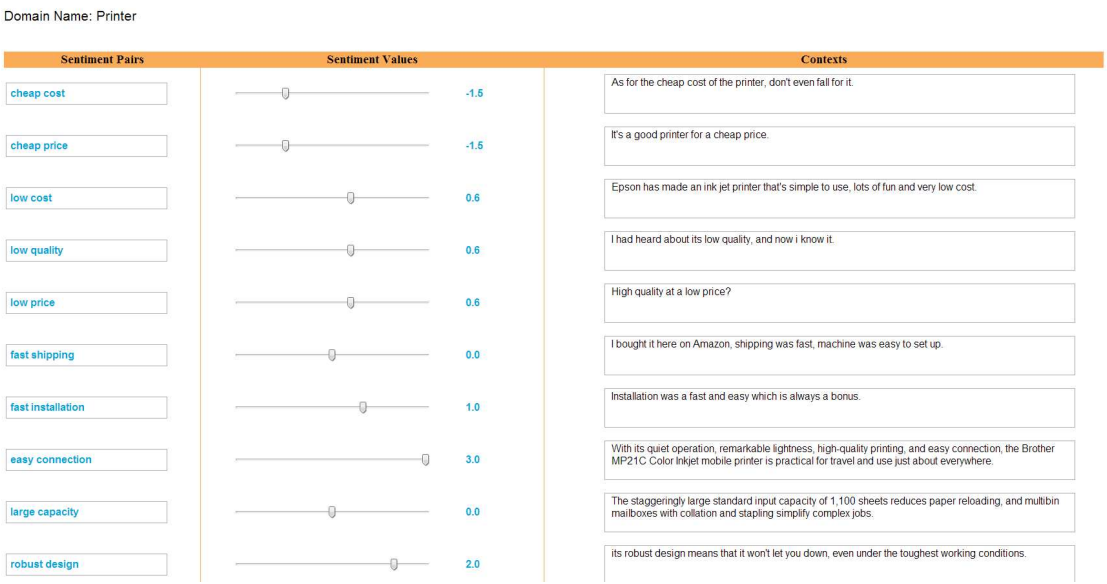


Figure 2.10: Text-based interface, sentiment pairs are ranked based on their frequency and presented in a list, each row contains a sentiment pair, a slider, and a sample sentence.

2.4.3 User Study

The evaluation methods for visualization systems was categorized into seven possible scenarios [152]. The guidelines for choosing the right scenario and performing the evaluation was also provided by Lam et al. Here we select the User Performance (UP) scenario because our purpose for performing the user study is to evaluate the effects of user supervision on the accuracy of the generated lexicon and compare the visual interface with the text-based interface measured by the users' performance. In addition, since we are interested in understanding user opinions about the visual interface and its success in supporting the intended tasks, the User Experience (UE) scenario is selected for this study as well. Based on the provided guidelines for the evaluation questions and methods by Lam et al., we designed the user study.

We asked 31 students in computer science to participate in this study. One of the participants barely interacted with the interfaces, both the visual and the text-based, due to her lack of interest in the study. She answered to the questionnaire but did not give any comments on the interfaces. Therefore, we had to ignore the results of this participant in all the analysis of this study. Each participant was given two separate tasks to perform with both the text-based and visual interfaces. In each task, participants were asked to adjust the polarity values predicted by the automatic algorithm. We also evaluate the visual interface through the quality of the generated lexicon and a questionnaire given to the participants at the end of the study. All the questionnaires asked in the study, including the approval letter from our university's ethics board are shown in Appendices A to D.

The way the study is conducted with regard to the different domains and interfaces is shown in Table 2.3. We used counterbalancing [73] in the design of our study, which means that to control the order effects, we separated the participants into two groups, each group interacts with both interfaces, but in a different order. In addition, we selected three products from the dataset, where each product is used with both interfaces as the first task as well as the second task. The participations' identification number, the products and the interfaces used in the first and second tasks are included in Table 2.3. For instance, our first five participants interacted with the text-based interface first and then performed the second task with the visual interface. Respectively, "Cell Phone" and "Camera" were the datasets for the first

Table 2.3: User study tasks and datasets

PIDs	Interface		Dataset	
	Task1	Task2	Task1	Task2
1-5	Text	Visual	Cell Phone	Camera
6-10	Visual	Text	Camera	Cell Phone
11-15	Text	Visual	MP3 Player	Cell Phone
16-20	Visual	Text	Cell Phone	MP3 Player
21-25	Text	Visual	Camera	MP3 Player
26-30	Visual	Text	MP3 Player	Camera

and second tasks. Different datasets are used in the first and second tasks to avoid the effect of becoming familiar with the data on the results.

2.4.4 Results

In order to evaluate user supervision effect on the quality of the generated lexicons, we calculated the accuracy of the lexicons before and after user supervision against the gold standard. To the best of our knowledge, there is no gold standard for context-specific lexicons; consequently, we constructed it for each domain in our dataset using the available score labels of aspects in the dataset(see Section 2.4.1). We extracted the sentiment words which describe these labeled aspects using the same set of rules given in Section 2.3.1. Then, having the polarity value of aspects and the sentiment words modifying those aspects, we construct the gold standard.

The average and standard deviation of the accuracy of the generated lexicons for different products and the number of pairs that were shown to the participants are shown in Table 2.4. The results indicate that user supervision improves the quality of the generated lexicons. However, in order to see whether this difference is significant or not, we ran statistical tests. Since we have a baseline for each product, which is the accuracy of the automatic algorithm before any user interaction, and there are two categories for the correct and incorrect sentiment pairs, we ran the one sample chi-square test (goodness-of-fit). In this test, we have an expected value for each category and the null hypothesis is that the results are not different from the expected values. When the p-value is less than 0.05, we reject the null hypothesis and conclude that the results are significantly different from the expected values or in this case, the baseline. In this study, we performed a separate test for each domain and interface. All the

Table 2.4: Average, standard deviation, and the χ^2 test of the sentiment lexicon accuracy before and after user supervision using text-based and visual interfaces

	Camera	Cell Phone	MP3 Player
Accuracy-U[a]	60.27	55.80	68.52
Accuracy+U[b], TUI[c]	91.04 (±4.61)	84.24 (±5.50)	88.91 (±3.49)
Accuracy+U, VUI[d]	94.21 (±2.21)	85.33 (±4.20)	91.10 (±3.08)
χ^2 test for T-UI	$\chi^2 = 36.444, p < 0.001$	$\chi^2 = 21.459, p < 0.001$	$\chi^2 = 16.598, p < 0.001$
χ^2 test for V-UI	$\chi^2 = 45.434, p < 0.001$	$\chi^2 = 24.225, p < 0.001$	$\chi^2 = 21.131, p < 0.001$
Number of pairs	138	114	141

[a] Accuracy before user supervision
[b] Accuracy after user supervision
[c] Text-based user interface
[d] Visual user interface

χ^2 and p-values in Table 2.4 show that user supervision significantly improves the quality of the lexicons. Now, to evaluate whether this improvement differs across the interfaces, we ran other statistical tests that are discussed in the next four sections.

Analysis of the Accuracy

In this section, we aim to determine whether there is a significant difference in the accuracy of the results between the interfaces. Therefore, the dependent variable is the accuracy of the lexicon, while independent variables are the type of the interface with two levels (text-based and visual), and the dataset with three levels (cell phone, camera, and MP3 player). Since we have more than one factor, ANOVA is a more suitable test than t-test. ANOVA has different versions for repeated measure and independent study designs and in order to select the best test for our study, we need to determine whether the factors of our study are between-subjects or within-subjects.

In our study, we have three levels for dataset and two levels for interface. Since each participant worked with both levels of interface, it is a within-subjects factor. However, each participant was involved in only two levels of the dataset out of three. Therefore, this factor is neither within-subjects, nor between-subjects. The reason that participants were not asked to complete all the tasks (i.e., 3 dataset * 2 interface = 6 tasks) is the limited time of the study for each participant. Consequently, participants were not involved with all the experimental conditions and the study cannot be considered as a repeated measure design [122]. Moreover, it is not an independent test (Factorial ANOVA), because each participant performed more than one task.

Table 2.5: ANOVA table for the accuracy of the first task

Source	Sum of squares	df	Mean square	F	P
Interface	116.033	1	116.033	4.473	**.045**
Dataset	280.080	2	140.040	5.399	**.012**
Interface * Dataset	20.059	2	10.030	.387	**.683**
Error	622.547	24	25.939		
Total	238430.260	30			

Table 2.6: ANOVA table for the accuracy of the second task

Source	Sum of squares	df	Mean square	F	P
Interface	1.019	1	1.019	.090	.767
Dataset	359.841	2	179.921	15.852	**.000**
Interface * Dataset	.104	2	.052	.005	**.995**
Error	272.394	24	11.350		
Total	239966.298	30			

However, if we consider the first and second tasks separately, we can apply a two-way ANOVA because different participants performed different experimental conditions and thus it can be considered as a between-subjects study. The results of applying the two-way ANOVA on the first and second tasks are presented in Tables 2.5 and 2.6, respectively.

The F ratio and the p-value for both factors, i.e. dataset and interface, are shown in Table 2.5. Since the p-value for the interface and dataset are less than 0.05, we conclude that in the first task, there are statistically significant main effects for both factors. Therefore, using the visual interface in the first task results in lexicons with higher quality and this difference is statistically significant. In addition, based on the analysis results, there is no statistically significant interaction between interface and dataset on accuracy, which means that the pattern of difference for the visual interface and text-based interface is the same for different domains. The same type of results for the second task of the study is presented in Table 2.6. However, it shows that for this task, no significant main effect was found for the interface. Although for the dataset, we still have statistically significant results. Similarly, there is no significant interaction between the two factors.

In summary, these two sets of results show that using the visual interface has a significant effect on the quality of the lexicon for the first task, but not for the second task. It is difficult to find a simple explanation for this difference, but one possible

reason can be the learning effect. For the second task, participants are more familiar with the task and already have some experience on how to perform it. Another possibility is the training instructions for each task. The purpose of the study and what participants should do during the study are explained at the beginning of the first task. The instructions given at the beginning of each task and their relative time length are shown in Table 2.7. This table indicates that although the total training time is similar for all participants, depending upon the order of the tasks, the pre-task instructions and their lengths are not the same for different groups of participants, i.e. participants who interacted with the visual interface in their first task versus participants who interacted with the text-based interface in their first task. This may be one of the reasons for having different observations in the first and second tasks.

Another point worth mentioning is that the type of user input is the same in both interfaces. We asked the participants to give us their input about the polarity of the sentiment pairs. Therefore, after the participants learn the task, they perform the polarity assignment task with a similar accuracy across the interfaces. Additionally, the main reason for providing the visual interface is having an easy to use interface by categorizing the aspects and providing proper user interactions. However, the results show that the visual interface has a significant effect on the quality of the lexicon in the first task. Therefore, it can be helpful for cases that the user is not familiar with the task. Since one iteration of this task for a domain creates a high quality lexicon, repetition of this task is not required. For instance, a product manager who wants to figure out the weaknesses of the newly released product from on-line reviews, or a website owner who writes about her expertise and wants to know whether people have positive or negative comments about her website can use our proposed method to generate the sentiment lexicon for their domains and use it to have a better sentiment analysis. Therefore, the fact that there may be a learning effect may not matter substantially in typical use cases.

Analysis of the Time Factor

In order to determine whether the visual interface helps the participants to complete their task faster or not, we recorded the amount of time that each participant spent to perform the tasks. The average and standard deviation of this value for each dataset

Table 2.7: Instructions for each task and their relative time length

	Purpose[a]	Length	Visual UI[b]	Length	Text UI[c]	Length	Total Length
First Task-Visual UI	Given	✓✓	Given	✓✓✓	Not-Given	×	✓✓✓✓✓
Second Task-Text UI	Not-Given	×	Not-Given	×	Given	✓	✓
First Task-Text UI	Given	✓✓	Not-Given	×	Given	✓	✓✓✓
Second Task-Visual UI	Not-Given	×	Given	✓✓✓	Not-Given	×	✓✓✓

[a]Instructions on what is the purpose of the study.
[b]Instructions on how to use the visual interface.
[c]Instructions on how to use the text-based interface.

and interface are shown in Table 2.8. On average, participants tended to finish their tasks faster using the visual interface. However, to see whether this difference is statistically significant, we applied a two-way ANOVA on the results. Similar to the analysis on the accuracy, we ran this analysis for each task separately. The results for the first and second tasks are presented in Tables 2.9 and 2.10, respectively. The p-values for the interface and dataset indicate that regarding the amount of time, no significant main effect was found for these factors. Therefore, there is no statistically significant difference between the amount of time for the visual and the text-based interface. Similarly there is no statistically significant interaction between interface and dataset.

Table 2.8: The average and standard deviation of the amount of time spent by the participants for each dataset (in seconds)

Interface	Camera	Cell Phone	MP3 Player
Text-based	635.8 (±237.15)	745.5 (±262.19)	745.2 (±290.26)
Visual	629 (±172.76)	737 (±315.19)	707.5 (±173.83)

It is worth mentioning that in this study, no instruction regarding time of completion were given and we did not ask participants to complete their tasks as quickly as possible. A potential study improvement would be to inform participants that the person who completes the study sooner than other participants and has the most accurate results will be given an extra bonus.

Analysis of the Post-Study Questionnaire

In addition to the accuracy and time that were recorded in this study, participants were asked to answer a questionnaire after they completed the tasks. The questionnaire is about the provided user interfaces. Some of the questions and the frequency

Table 2.9: ANOVA table for the amount of time spent by the participants for the first task (in seconds)

Source	Sum of squares	df	Mean square	F	P
Interface	20750.700	1	20750.700	.262	**.613**
Dataset	168283.400	2	84141.700	1.064	**.361**
Interface * Dataset	18963.800	2	9481.900	.120	**.888**
Error	1898476.800	24	79103.200		
Total	21930667.00	30			

Table 2.10: ANOVA table for the amount of time spent by the participants for the second task (in seconds)

Source	Sum of squares	df	Mean square	F	P
Interface	27300.833	1	27300.833	.669	**.422**
Dataset	6301.267	2	3150.633	.077	**.926**
Interface * Dataset	15388.867	2	7694.433	.189	**.829**
Error	979640.000	24	40818.333		
Total	12141667.000	30			

of their answers are shown in Fig. 2.11. We mapped the answers to an ordinal scale from 1 to 5, where 5 is the best. The stacked bar charts present the frequency of these scales for each question. Questions 4 and 5 ask participants to indicate how easy the interfaces are to use. Therefore, these questions aim at comparing interfaces and the participants' answers indicate which interface is easier to use. Similarly, questions 6 and 7 ask how helpful the interfaces are in the polarity assignment task. It is clear in Fig. 2.11 that more participants assigned the highest score to the visual interface than to the text-based interface. In overall, participants assigned higher scores to the visual interface rather than the text-based one. However, in order to have a better analysis, we ran the Wilcoxon signed-rank test. Since in this case, our dependent variable is ordinal, we used a non-parametric statistical test.

The results of the Wilcoxon signed-rank test for questions 4 and 5 show that there is significant difference between the participants' rating for the visual and the text-based interface ($Z = -2.514, P = 0.012 < 0.05$). Similarly for questions 6 and 7, the scores of the visual interface are significantly higher than the text-based interface ($Z = -3.251, P = 0.001 < 0.05$). In addition, we asked this question "Overall, which user interface do you prefer to work with?". 25 participants answered visual interface, while 5 stated that they prefer to work with the text-based interface.

Additionally, the frequencies of the answers to questions 11 and 19 indicate that most of the participants think that the tree cloud makes the task easier and also the layout of the visual components is appealing.

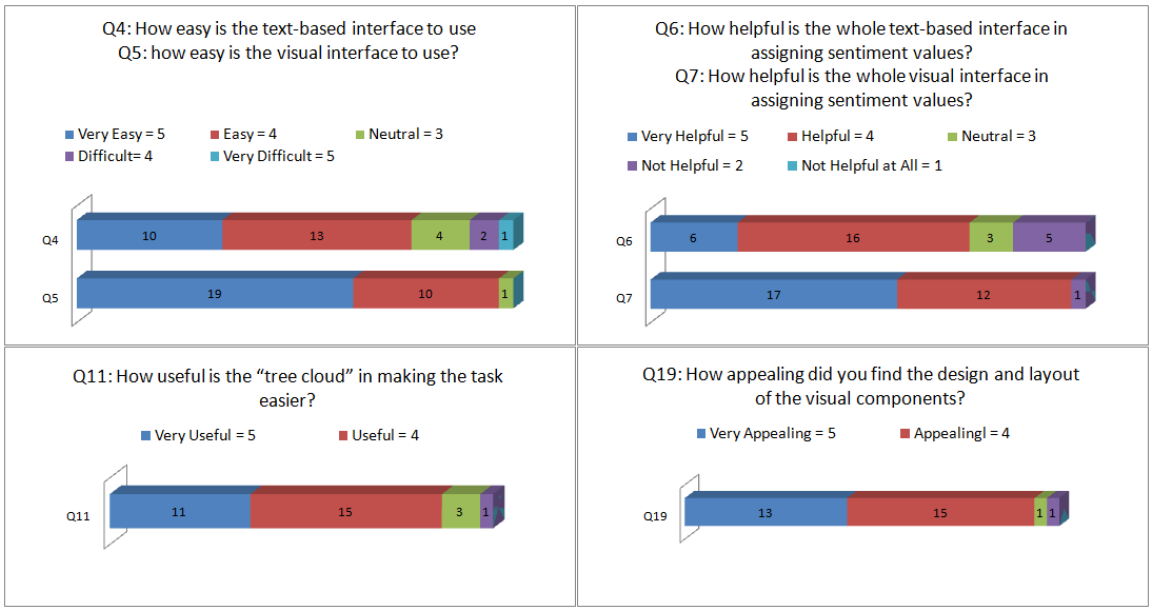


Figure 2.11: Stacked bar charts showing the frequency of the participants' answers to a number of questions.

Participants' Comments on the Interfaces

Participants' comments on the provided interfaces that were given at the end of the study are discussed in this section. Participants give responses such as *"For the visual system I like the tree a lot, felt well organized"* (participant ID = 12), *"The text based system was harder to follow, scrolling made it hard and easy to lose your place"* (participant ID = 27), and *"The text-based interface, it is relatively easy to use and I think it will bring more accurate information"* (participant ID = 5).

Since it is not possible to present all the comments, we performed a qualitative analysis on their descriptive answers. Our method follows a diversity analysis (coding and analysis of patterns), combined with simple frequency descriptions. This approach is derived from Jansen [128]. In this analysis we looked at the total number of 30 open text answers. These answers are processed to extract coded concepts. The concepts are then organized under dimensions and sub-dimensions. This constructs

a tree structure of the coded data (Fig. 2.12).

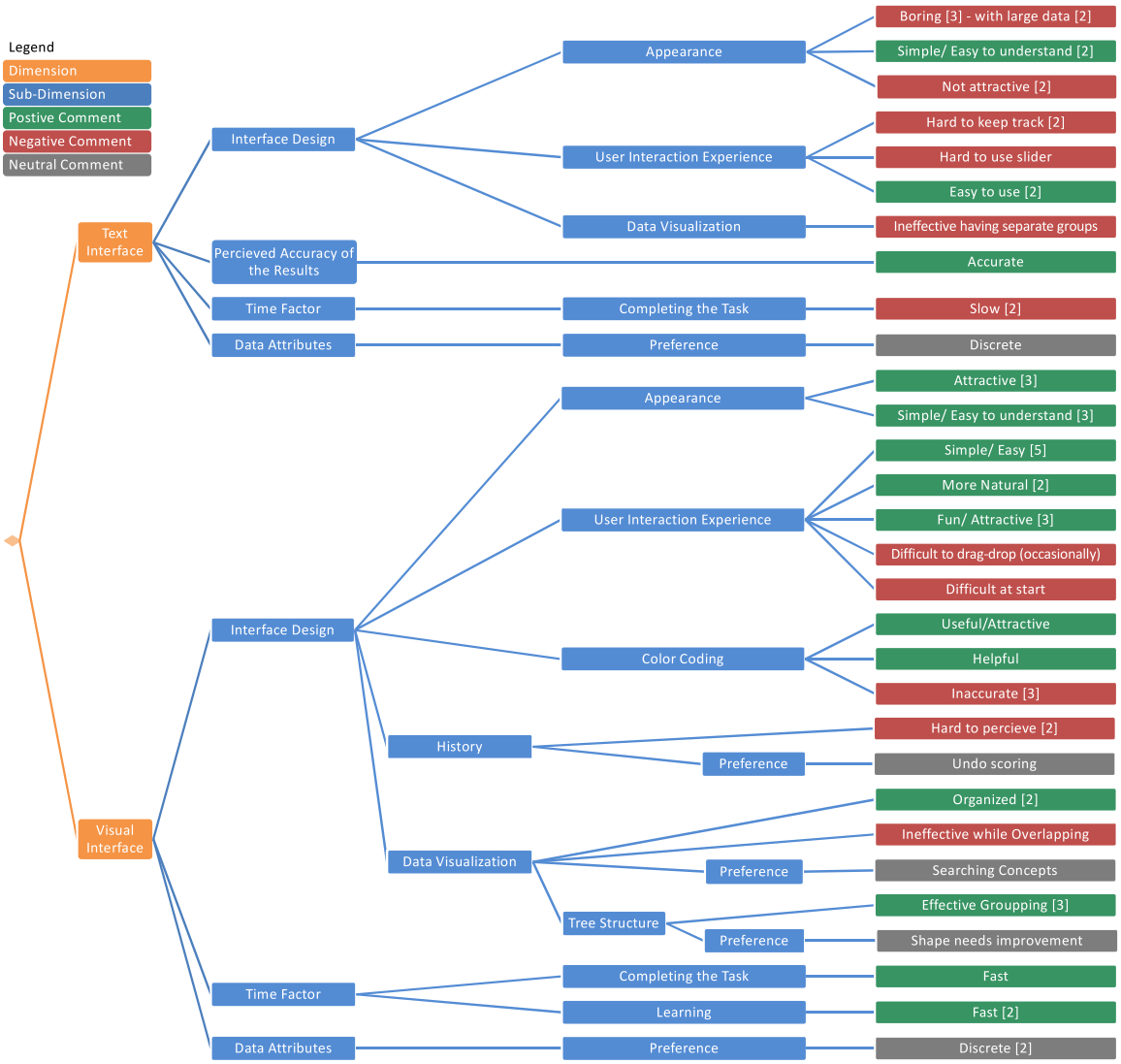


Figure 2.12: Concept map resulting from qualitative analysis of open text answers. The tree divides in two main dimensions: Visual and Text. Major sub-dimensions are interface design and time factors. In this figure frequencies of observation are added in brackets. Concepts without brackets represent one comment.

By following the patterns in the concept map, it is noticeable that a large portion of comments focus on interface design. Overall, the appearance and user interaction of the visual interface is commented about more positively than the text interface (attractive and natural interaction comparing to boring and not attractive). However, difficulties were mentioned regarding interaction with both interfaces; text-based is hard to keep track, or as is using the slider, while it was mentioned once that the

drag and drop approach in the visual interface is difficult.

Data visualization is regarded as more effective and organized (considering the tree cloud). There are also few preferences stated concerning a concept search function, and improvements to the tree shape. Additionally, there are both positive and negative comments regarding visual-based color coding and history of the user interactions (i.e. dimming the tree branches that have been inspected, polarity color in the polarity assignment view). The visual interface is perceived to be faster, both in learning the interface and when completing the task. Finally, there are preferences for changing the polarity scores to discrete values in both visual and text-based interfaces, since it is easier to interpret. Overall the visual interface received a greater number of comments compared to the text-based interface. It also has a higher frequency of positive comments than the text-based interface ($Visual : \frac{26}{38} = 68\%; Text - Based : \frac{5}{16} = 31\%$).

Moreover, we asked participants to tell us how to improve the visual interface. Four participants said they would like to see the topic of the branches (i.e., the categories of aspects) in the tree cloud since it would help them in determining these categories. Three participants declared that in the polarity assignment view, they would like to see a difference between the sentiment pair(s) that have already been moved and the ones they still need to review.

2.5 Conclusions

We combined visualization techniques and automatic methods for generating context-specific sentiment lexicons. We proposed a novel interactive visualization for involving the user in the process of polarity assignment for the first time. To evaluate the generated lexicon, we ran a user study with thirty participants. The results demonstrated the extent to which the engagement of the user in the polarity assignment improves the quality of the lexicon and that this improvement is statistically significant. Furthermore, we discussed in Subsection "Analysis of the Accuracy" that using the visual interface in the first task significantly improves the quality of the lexicon, while this improvement cannot be observed in the results of the second task. However, this difference does not disclaim that the visual interface is an easy to use interface that facilitates the user interactions and enhances her experience.

Analysis of the time factor showed that there is no significant difference between

the amount of time that participants need to complete the tasks using different interfaces. Although the visual interface is not significantly faster than the text-based interface, most of the participants preferred to use the visual interface as they believe it is easier to use and its components are more helpful in the polarity assignment task. Besides, the visual interface can be useful in cases that the user does not have any experience. We note that polarity assignment for large lexicons is a very tedious task and we speculate that offering an improved interface that users prefer would likely reduce fatigue and user disinterest.

2.5.1 Limitations and Future Work

There are several future research directions which can be pursued to extend the work in this chapter. We list them below.

1. Evaluating the effect of user supervision on improving the sentiment pair extraction. In this chapter, we involved the user in the polarity assignment process and evaluated the accuracy of the assigned polarities. One aspect that could have been evaluated is how much user involvement enhances the quality of extracted sentiment pairs. We would have liked to add user interactions that enable users to accept or reject the extracted aspects or even add new aspects that are missed.

2. Polarity assignment for aspects. Besides sentiment words, aspects can also imply opinions [305]. For instance, "noise" or "constant delay" are nouns and noun phrases that imply negative opinions themselves. Adding user interactions that make it possible to assign polarity to the aspects can be helpful in this case.

3. Evaluating the generated sentiment lexicon by using it in sentiment analysis applications. Since we needed a gold standard to evaluate the generated lexicon, we chose a fairly small dataset. Another way to evaluate the generated lexicon is to use it in an application such as sentiment classification and see how much it improves the results of the classifier. However, this type of evaluation may add errors of the classifier to the results. Using larger datasets and evaluating the generated lexicon in this way is also considered as future work.

4. Taking into account that the polarity of sentiment words may change over time. Since the dataset used in the experiments of this chapter is from 2004, the meaning of some sentiment words may have changed, especially for datasets related to technology. For instance, if having a blue screen was a positive feature for MP3 player at that time, it can be different now with existing touch screens. Another example is having 64 MB memory for a cell phone 10 years ago versus today. It was a good feature 10 years ago, but now having this capacity of memory is not acceptable. Even for datasets that have been produced recently, this change of the meaning of numbers may happen for different aspects. For instance, 45 minutes waiting time in the customer service line is negative, while if this amount of time is needed to fully recharge the cell phone, it is positive. Identifying the sentiment words with a change in their meaning over time would be an addition to the proposed system.

5. Considering polysemous sentiment words. One of the limitations of our proposed method is not considering the ambiguity of some sentiment words. For instance, the sentiment pair "(cheap, battery)" can be positive or negative depending on the meaning of the sentiment word, "cheap". Except the sample context of the sentiment pair shown in both interfaces, no other information is given to the user that can help her in disambiguating the sentiment pair. Clarifying the meaning of the ambiguous sentiment words should be added to the system as future work.

6. Assigning polarity to phrases. Another limitation of our work relates to the polarity prediction module of the automatic algorithm when dealing with phrases instead of single words. For instance, in the phrase "this camera is easy to use", "easy to use" is the adjective phrase. As explained, if the automatic algorithm does not find any evidence to predict the polarity of the sentiment pair, it considers the polarity of the sentiment word (i.e., easy) in the general purpose sentiment dictionary as the predicted polarity for this pair. However, it is clear that the sentiment of a phrase can be different from the sentiment value of its head, which is the word that determines the syntactic type of the phrase. For example, in the context "this cell phone is easy to break" , the sentiment phrase,

"easy to break" has a negative polarity, while the sentiment of the head of the phrase,"easy", is positive. Adding some techniques to calculate the polarity of the adjective phrases can improve the accuracy of the automatic algorithm.

7. Taking into account negation when analyzing sentiment. We did not consider negation in the reviews for polarity calculation (Eq. 2.1). Using NLP techniques for identifying negation and considering it in the polarity assignment would increase the accuracy of the assigned polarities to the sentiment pairs.

8. Updating the algorithm based on the user feedback. In this chapter, the user input is not considered as a feedback to the automatic algorithm. Therefore, changing the polarity of a sentiment pair does not have any effect on the polarity of other sentiment pairs. Considering the user input about each sentiment pair as a feedback, recalculating the polarity of other sentiment pairs, and updating the visualization based on the new predicted polarities may reduce user effort as well.

Chapter 3

Microblog Retrieval for Parliamentary Debates

3.1 Introduction

Microblog messages are another type of online user-generated content with particular characteristics. The volume of these messages is increasing significantly with the popularity of microblogging services. Twitter is one of the most popular microblogging platforms with more than 310 million active users [271] who post text-based messages of up to 140 characters. These posts are known as tweets and contain information of broad interest [66]. People write about their opinions, experiences, ideas and feelings on Twitter every day. They also spread news about important world events widely on Twitter [308, 307].

This rapid growth of social networks' popularity makes them significant sources of information. Different types of applications such as event detection [246], news story identification [220, 59], recommendation systems [41] and opinion mining [202] can benefit from the information embedded in the content of these messages. As in most social communication media, politics is a frequent conversation topic. Politicians, party organizers, the press media and the general public use social media to express opinions, compete for public attention and recruit new supporters [6, 99, 294]. As a result, there is a genuine interest in mining this diversely rich and endless source of public opinions. For instance, Twitter was extensively used for political deliberation [269] during live media events such as presidential debates [250] and as a platform for expressing opinions about presidential candidates [49].

In national legislatures, government laws are introduced, debated and voted by the members of the parliament or congress. Recently established open government data policies render this information readily available. At the same time Twitter has become widely adopted by the public as a platform for engaging in discussions about the various bills under debate in the parliament. While several computational tools for the textual analysis of social media data have been proposed, tools for retrieving such

data effectively are still lacking [163]. The method proposed in this chapter addresses this gap. Our specific goal is to retrieve Twitter posts related to political debates held in the parliament, and associate them to the specific debate they refer to. Such association is important because it allows identifying public opinion about political decisions and bills under debate. Yet, it must be accurate, so that retrieved content is relevant, and comprehensive to ensure that diversely expressed opinions are captured. Politicians, political analysts and journalists can benefit from this association as they may want to follow specific debates/topics and need to distinguish what people are saying about their debates of interest.

A well-recognized challenge when working with tweets is their textual content being short and riddled with acronyms, slang, incorrect spelling and grammar [238]. Adding to this, our research problem poses its own specific challenges. One factor that renders it particularly difficult is the semantic relatedness among debates. Besides all target tweets having "federal politics" as their underlying topic, different debates can have other topics in common. For instance, during the second week of May 2014 debates on the issues of "Kidnapping of Girls in Nigeria" and of "Aboriginal Affairs" were taking place in Canada. Although they are distinct, both deal with similar issues of violence against women, which makes the association task especially challenging. Most contributions reported so far on tweet classification or retrieval consider well-differentiated categories such as sports, politics, economics and technology [153, 257], which constitutes a scenario simpler than the one considered here.

An additional challenge for our problem is that the distribution of tweets over debates is severely imbalanced. Some debates may raise issues that are deemed important to a larger audience and consequently they will originate more tweets than other debates related to more specific questions. Moreover, the relevant tweets to any debate are a small fraction of the overall volume of data. Another issue is the dynamic nature of the problem, as new debates are continuously introduced whenever a legislative session is held. It would not be suitable to train a model on old data, but it would also be infeasible to label large quantities of data every time new debates start to be monitored.

3.1.1 Research Problem

We define our research problem as follows:

Definition 2 *Given a set of tweets* $T = \{t_1, t_2, ..., t_n\}$ *and a set of debates* $D = \{d_1, d_2, ..., d_m\}$*, the task is to retrieve, for each debate* $d_i \in D$*, a set of tweets* $T_i \subseteq T$ *so that tweets in* T_i *are relevant to the debate* d_i*, and* $T_i \cap T_j = \varnothing$ *for* $i \neq j$*. Equivalently, we define for any tweet* $t_k \in T$ *a function* $f(t_k) = d_i$ *if the retrieval method associates* t_k *with* d_i *or* $f(t_k) = \varnothing$ *if* t_k *is not retrieved, and therefore considered as "non-relevant" to any of the debates.*

Consistent with a typical information retrieval setting, we assume labeled data is not available for training a method. However, we extend this setting assuming an external information source may be accessed that can manually label a given tweet with any or none of the debates in D. We also assume that there is a practical cost of accessing the information source, so it is important to minimize the number of instances to be presented for feedback. Note from the definition that we presume each tweet is related to at most one debate.[1]

Considering that no labeled data is available to train a model, our approach relies on two aspects: (1) inspired by the idea of pseudo-relevance feedback [292], we refine the queries by which tweets are retrieved to maximally improve recall and precision and (2) we involve the user in the retrieval process so that her domain knowledge can be leveraged. Consistent with our objectives in this thesis, we first propose Active Tweet Retrieval (ATR) [178], which employs multiple active learning strategies for increasing the benefits from user supervision by selecting the most appropriate labeling requests. As opposed to other active retrieval approaches, the set of strategies introduced here exploit particular features of Twitter for selecting the labeling requests. To further facilitate the association process, we introduce an interactive and exploratory tool, named ATR-Vis (Active Tweet Retrieval Visualization)[2]. This tool visualizes the tweets that have already been associated with each debate, and also

[1] This can be also stated as a classification problem, where we try to assign each tweet to a debate (or assign it to an additional "non-relevant" class when it is not retrieved). However, since initially there is no labeled data and queries are inferred from the debates, the problem is more naturally posed as an information retrieval problem.

[2] This work was done in collaboration with Eder Carvalho, Maria Cristina Ferreira De Oliveira and Rosane Minghim from Universidade de São Paulo, Brazil, and published as [177].

those deemed as important for the user to inspect and assign. Yet, the user is also offered other visual means to explore the data, so that she can associate tweets based on this exploration and beyond what it is being "suggested" by the system. While it relies on user involvement to improve the association of tweets, ATR-Vis aims at keeping this involvement to a minimum.

Our main contributions in this chapter may be summarized as follows:

- The proposal of a set of active retrieval strategies that are specific to Twitter and increase the retrieval accuracy in terms of precision and recall while minimizing user effort, i.e. the number of labeling requests.
- A comparison of the proposed approach with a state-of-the-art active retrieval method and its evaluation on different datasets.
- The presentation of an interactive framework, ATR-Vis, that enables non-technical users to employ the aforementioned active learning strategies while exploring the space of potential tweets and gaining a better understanding of the results.

3.1.2 Overview

This chapter discusses the problem of associating microblog messages with a set of predefined topics. We present ATR-Vis, a user-driven visual approach for the retrieval of Twitter content. We start by proposing a method that finds the most relevant topic to a given tweet. Adopting the proposed framework in Chapter 1, we introduce a set of novel active learning strategies based on specific features of Twitter to involve an analyst in such a way that a major improvement in retrieval coverage and precision is attained with minimal user effort. In addition, to enable non-technical users to benefit from the aforementioned active learning strategies, visual aids are provided to facilitate the requested supervision. This supports the exploration of the space of potentially relevant tweets, and affords a better understanding of the retrieval results. We evaluate our proposed retrieval method on two datasets related to parliamentary debates. Quantitative results show that our approach achieves high retrieval quality with a modest amount of supervision. In addition, our interactive visualization is also evaluated with three domain experts. Finally, several use cases illustrate the functionality of ATR-Vis in different scenarios.

This chapter is organized as follows. A survey of related studies, which includes microblog processing approaches and exploratory tools for Twitter is discussed in Section 3.2. Our proposed method is described in Section 3.3 and its evaluation in Section 3.4, covering quantitative experiments, use cases illustrating the applicability of our proposed framework and a qualitative analysis conducted with potential end users. Section 3.5 presents concluding remarks and future research directions. Besides the textual content of a tweet, there are other pieces of information that can be extracted from them, e.g. the author of the tweet or the time and date that the tweet was published. The list of tweet fields that we refer to in this thesis and their descriptions is summarized in Appendix E.

3.2 Related Work

We first explain related work on processing content of tweets and then we present studies performed on visual analytics for Twitter analysis.

3.2.1 Processing Microblog Content

Given the importance of microblog content and the multiple challenges described in Section 3.1, several research efforts have addressed Twitter content retrieval, classification and/or analysis. Initial attempts relied on manual identification of relevant keywords that are used to filter relevant posts either as part of the Twitter API parameters [22, 80] or in a postprocessing step [48, 237]. However, due to the noisy and evolving nature of tweet terminology, it is hard to ensure a proper recall, i.e. capturing all relevant topics without biasing the results towards certain specific keywords [74].

Several studies tried to use hashtags for finding relevant tweets. Since hashtags are good descriptors of the content of a tweet, they can be used to facilitate searching. However, there are a few problems that prevent hashtags from being reliable features for retrieving and categorizing tweets. For instance, the majority of tweets do not contain hashtags. It has been stated that only about 8% of tweets contain a hashtag [92]. However, this number varies between different datasets. One of the reasons that the majority of tweets do not contain any hashtags is that many users are not aware of existing hashtags when they publish their tweets. In this case, hashtag recommendation systems can be of important value [92]. In addition, without hashtag

recommendations, users may choose different hashtags to indicate the same concepts [71], which creates a long list of infrequent hashtags. Another problem is that some hashtags are very general and it is difficult to assign them to a specific topic. For instance, hashtag "#fb" is automatically added by some Facebook [68] applications that share users' posts on Twitter [12]. Moreover, it has been discussed in a study that Twitter users include hashtags in their tweets for a conversational function to promote particular topics rather than organizational purposes [118]. Therefore, although it is important to consider hashtags as informative entities, they cannot be used alone to classify tweets.

Other approaches considered query expansion to enrich the query terms and overcome the vocabulary mismatch problem [184, 101]. More recent methods resorted to external knowledge sources such as Wikipedia [288], Freebase [95], and WordNet [223] to obtain additional related terms to expand the query [224, 171]. However, expanding queries can undermine the precision of the retrieval as more generic terms are included [191]. Our query expansion approach aims at adding features while preventing a loss in retrieval precision.

To alleviate the short and noisy nature of tweets, a group of methods utilize external auxiliary datasets to augment the information related to tweets [218]. For instance, tweets can be linked to Wikipedia articles (concepts), which are rich sources of information. Using Wikipedia to automatically enrich a text has been studied in different tasks such as word sense disambiguation [187]. Similarly, this technique has been used to augment information of entities in tweets in order to improve tweet classification [86, 114]. However, finding relevant Wikipedia concepts of tweets is itself not an easy task. To associate a tweet to a Wikipedia page, some methods simply check for pages that are dedicated to the words appearing in the tweet and construct a list of candidate pages [88]. Then, they rank the candidate pages by the frequency of those words. However, since people use many variants for the same entity in Twitter [166], many of the words cannot be found in Wikipedia. More advanced methods for associating tweet entities to Wikipedia concepts can be found in [185, 165]. A semi-supervised approach based on graph regularization has been introduced in [117] that simultaneously considers mention detection and disambiguation in the tweet

wikification[3] task.

Similar to our work, other studies highlighted the advantages of considering the structural information surrounding a tweet such as the hashtags, URLs and replies to other posts [172, 173]. To identify discussed topics by members of the Dutch Parliament on Twitter, Kalmeijer clustered hashtags extracted from their tweets, and argued that the quality of the generated clusters are fairly good but the identified topics are sometimes too specific to be used in political research [134]. Hashtags also been used for topical clustering and classification of tweets [238]. In addition, it has been shown that sometimes expanding the URLs present in tweets by retrieving the text of their target documents has a negative effect on the performance of the classifier [238]. Other researchers classified trending topics into general categories using the network structure of Twitter. For example, network-based classification techniques have been compared with bag-of-words methods for the task of classifying tweets [153], which demonstrated that using the Twitter network structure improves the accuracy of the classification significantly.

Most of the reported studies use meta-data or other knowledge sources along with the textual data in their proposed methods. However, there are a few methods that only use the tweet message to detect tweets related to events. For instance, a semi-automatic approach that applies heuristic rules to detect tweets related to social events has been introduced [124]. Heuristic rules are used to detect the presence of time, persons or locations in the tweets published by event broadcasters. If these aspects are present, the tweet is considered as an event-related tweet. Otherwise, it is not related to social events. Therefore, a training set can be constructed by automatically classifying tweets published by event broadcasters, which is used to train a Naïve Bayes classifier for classifying other tweets [124].

A number of researchers focus on using topic models to classify tweets [60, 110, 218, 189, 297, 138]. An interesting work on separating two types of tweets related to public events is reported by Hu et al. [116]. In their classification, episodic tweets refer to a specific topic of the event and steady tweets are related to the general theme of the event. They propose a joint statistical topic model to identify these types of tweets.

[3]Wikification is the term used to refer to the process of entity linking to Wikipedia articles.

The labeling of Twitter data is an expensive process with its usefulness and generality limited to a certain thematic context and time window. Therefore, several studies have focused on learning models with limited labeled data. Semi-supervised techniques rely on large amounts of available unlabeled data along with a small amount of labeled data. For instance, a semi-supervised Bayesian network model for Twitter topical classification was proposed in [43], and a semi-supervised SVMRank for ranking tweets based on their credibility is described in [100].

Similarly, active learning is a special case of semi-supervision where the method itself requests instances to be labeled from an information source. The use of active learning techniques for analyzing microblog messages is a relatively new research topic. The importance of considering social relations and user similarity among microblog posters in selecting the instances to be labeled for the task of topical classification was shown by Hu et al. [115]. Another recent work used crowdsourcing to label selected tweets for identifying informative tweets for disaster management [125]. Uncertainty sampling is a common active learning technique that has been used for selecting labeling requests. For instance, this technique was used for named-entity disambiguation in Twitter [215], where a Naïve Bayes classifier was trained and the most uncertain samples are selected to be labeled in order to improve the performance of the classifier. The methods adopted in the latter two works consider only the textual content of tweets in training their classifier and selecting the instances for labeling requests. They do not explicitly model other specific Twitter features, e.g. reply-related information. Moreover, none of the previous approaches considered this task within an interactive exploratory setting, where a human user can participate in the labeling task.

Active retrieval would be the analog of active learning for the task of information retrieval [127]. Similar to active learning, the retrieval system is allowed to request instances to be labeled on an interactive basis for the sake of improving retrieval precision or recall. Again, it is assumed that there is a cost associated with each labeling request, so the number of requests should be minimized. Two major differences are that active retrieval is expected to face a *cold start*, i.e. starting with no labeled data, and some sort of query is available representing the information need. ReQ-ReC [155] can be arguably considered as one of the state-of-the art active retrieval systems. The

method consists of a sequence of two cycles. The inner cycle aims to improve the precision of the retrieval by training an SVM classifier, where uncertain instances are selected for user labeling. The outer cycle aims to improve the recall by automatically formulating a new query that is obtained from low-ranked but relevant documents in the hope of increasing the diversity of the retrieval. While ReQ-ReC does not provide any exploratory features, we adopt it as a basis for comparing the back-end of our approach.

3.2.2 Visual Analytics for Microblog Content

Twitter has been attracting considerable attention as a data platform for visual text analytics. Initially, the tools focused mainly on providing platforms for visualizing Twitter content, users and images beyond the conventional list layout [203, 62, 59, 181]. Most tools apply dynamic topic models and present them using ThemeRiver-inspired visualizations [107] to convey a summary of the conversations over time. A second generation of systems took the idea of extracting topics over time and applied more advanced text processing algorithms to improve content analysis. For instance, Leadline [63] aims at extracting the major events that led to changes in the topical themes and characterizing each event with information on who, what, when, and where. The analysis of evolving topics from the perspective of how they compete among each other and how they diffuse to different users were addressed in [294, 164, 258]. Yet other studies proposed methods for predicting revenue or stock prices from Web data. A visual analytics system to predict the box-office success of a movie was proposed by Lu et al. [169], where the number of mentions in Twitter per day is one of the few variables of the model. Retrieval is restricted to the hashtag posted by the movie's official Twitter account. However, none of these previous works consider a systematic strategy for accurate retrieval of relevant tweets, rather they analyze all Twitter posts that match a given set of keywords.

Solutions such as SensePlace2 [174] and the system by Chae et al. [38] focus on providing geo-visual analytics for understanding place, time, and theme components of evolving situations. Such solutions have been proved useful to improve situational awareness in monitoring catastrophes based on their reporting on Twitter. Scatterblogs2 [23] is another visual analytics tool for situational awareness, but unlike the

previous two it supports the expansion of an initial manual query by looking at the co-occurrence of tweets retrieved with the first query. These filtering aids are mostly based on textual content only, without taking Twitter-specific features into account.

Concerning the target goal of improving retrieval capability, the recent tool introduced by Liu et al. [163] is probably the closest to the approach proposed in this chapter. It also exploits the specific characteristics of microblog data to improve retrieval performance when looking for information in twitter posts. The authors propose an uncertainty-aware microblog retrieval model to quickly retrieve salient items, i.e. posts, users and hashtags, and provide an estimate of the uncertainty associated with the retrieval model. The retrieval relies on a previously introduced uncertainty-based mutual reinforcement graph model, in which the quality of a post content, user social influence and hashtag popularity mutually reinforce each other in order to determine the relevance of a post to a query. A composite visualization using a graph metaphor is proposed to support analysts to understand the retrieved data and interactively refine the retrieval model.

Both our work and Liu et al.'s aim at providing visual means for improving and facilitating retrieval. However, Liu et al. focus on the authorativeness and popularity of the posts, while we focus on their thematic relevance and the ability to identify as many relevant posts as possible. Although Liu et al. acknowledge the difficulties associated with evaluating recall, its estimation is highly important for our motivation. In addition, we consider the simultaneous retrieval of multiple, unbalanced and closely-related topics, which makes it challenging to achieve a high precision retrieval. We address these challenges by proposing active retrieval strategies, where a user provides feedback on request by the system—in addition to the feedback from her own exploration—to improve retrieval results.

3.3 Proposed Method

The main goal of our proposed approach is, given a set of target debates, to retrieve tweets relevant to each debate, attempting to maximize precision and recall. Our framework for handling the problem has three major components. The first component is responsible for the initial unsupervised retrieval, trying to achieve the best possible response without any human intervention. Clearly, the better this component

performs, the less load is put on a user in subsequent steps. Retrieved tweets provide a pseudo-relevance feedback [292] that is used to improve the set of discriminative features. This is in agreement with our proposed generic framework for interactive text analytics of user-generated data in Chapter 1, as we use predicted labels of tweets for improving the model.

The second component encompasses a set of strategies introduced to involve the user in those critical cases where her involvement is likely to yield an increase in the retrieval precision or recall. This component is tightly connected with the third one, i.e. the interactive visualization component. These components present selected instances for manual labeling and enable user exploration over the collection of tweets and their characterizing features. The results of user interaction are fed back into the retrieval engine, which analyzes the given information to automatically extract new discriminative features that will guide further iterations of the retrieval process. The whole process is illustrated in Fig. 3.1, while Table 3.1 summarizes the notation used throughout this chapter.

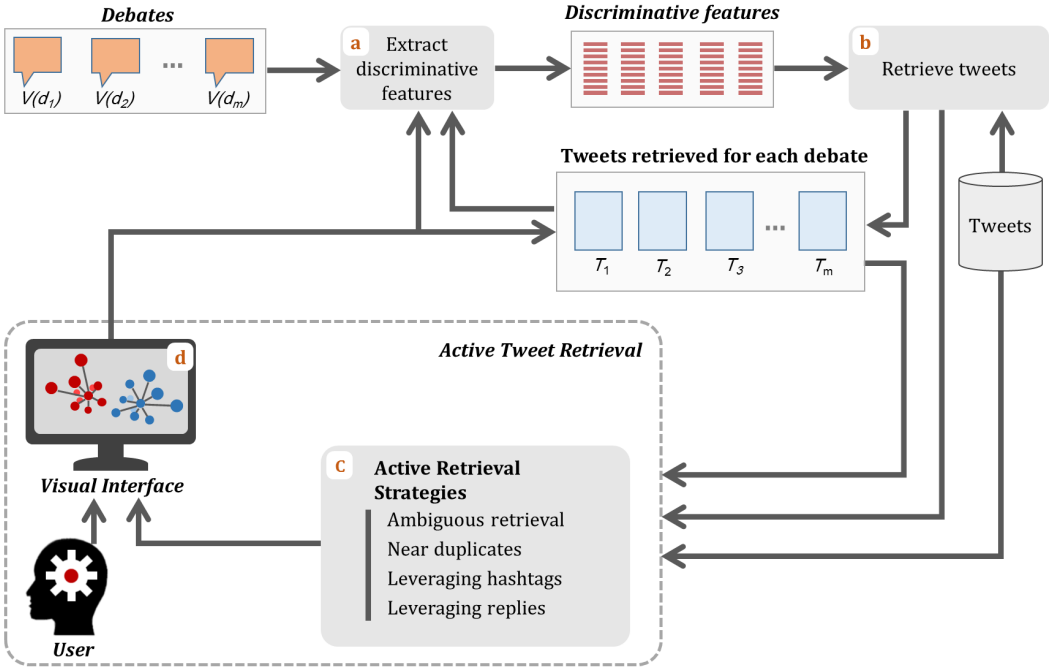


Figure 3.1: The proposed framework for retrieving tweets relevant to a set of political debates. The unsupervised retrieval consists of extracting discriminative features (a) and retrieving tweets (b), and the active retrieval component selects the labeling requests (c) and updates the retrieval model based on the obtained labels (d).

Table 3.1: Notation used in this chapter

Notation	Description
T	Set of tweets in our dataset
t_i	A tweet in T
D	Set of relevant debates
d_j	A debate in D
$\mathbf{k}$	List of all extracted discriminative features of different debates
k_p	A discriminative feature in $\mathbf{k}$
Ω	Matrix of weights $\omega_{i,j}$ representing the importance of feature k_i to debate d_j
H	Set of all hashtags occurring in T
h_o	A hashtag occurring in T
$f(t_i)$	Returns the debate associated with tweet t_i by the retrieval method
T_j	Set of tweets that are associated with debate d_j by the retrieval method
$F(T')$	Returns the frequency distribution of debates associated with a set of tweets T' by the retrieval method
$T(h_o)$	Set of tweets having hashtag h_o
$R(t_i)$	Set of tweets in the reply chain of tweet t_i
$df(h_o)$	Debate frequency of hashtag h_o
$V(T')$	Virtual document constructed by concatenating the textual content of a set of tweets, T'
$sim(t_i, d_j)$	Similarity value between tweet t_i and debate d_j

3.3.1 Unsupervised Tweet Retrieval

In order to extract a set of keyterms for retrieving relevant tweets, we follow an approach similar to the one introduced by Golestan Far et al. [93], which generates a query by extracting discriminative terms from a document representative of the information need. We produce a representative document for each debate by concatenating all transcripts of a single debate, which can span multiple parliament sessions and days. From each document a set of discriminative keyterms, i.e. terms with the highest tf-idf values, is extracted for its corresponding debate. All these discriminative features are added to a list of features $\mathbf{k}$, which is used for retrieving relevant tweets from T. Therefore, we define the list of discriminative features as follows:

Definition 3 *Let* $\mathbf{k} = (k_1, k_2, \ldots, k_s)$ *be the list of features, and let matrix* $\Omega = (\omega_{i,j}) \in \mathbb{R}^{s \times m}$ *indicate the importance of each of the* s *features in each of the* m *distinct debates.*

Following the above definition, a matrix component $\omega_{i,j}$ contains a non-zero value if feature k_i is selected as a discriminative feature of debate d_j. A debate d_j is thus represented as a feature vector $\mathbf{d_j} = (\omega_{1,j}, \omega_{2,j}, \ldots, \omega_{s,j})$, while for a tweet we define its feature vector as:

$$\mathbf{t_i} = (\alpha_{1,i}, \alpha_{2,i}, \ldots, \alpha_{s,i}), \alpha_{p,i} = \begin{cases} 1, k_p \in t_i \\ 0, k_p \notin t_i \end{cases}, p \in [1, \ldots, s]. \tag{3.1}$$

Similarity between a debate d_j and a tweet t_i is given by the dot product of their feature vectors, i.e. $sim(t_i, d_j) = \mathbf{t_i} \cdot \mathbf{d_j}$. When retrieving tweets we calculate the similarity score for each tweet-debate pair, and the tweet is assigned to the debate for which the similarity score is the highest provided this similarity is above a certain threshold. Details about the setting of this threshold are provided in Section 3.4.2.

Using just the debate transcripts as a source for the queries is not enough, as there is a potential mismatch between the formal vocabulary used in the debates and the informal one adopted in Twitter. Therefore, inspired by the pseudo-relevance feedback approach [292] we use the retrieved tweets to further expand the list of discriminative keyterms. In addition to regular terms, discriminative Twitter-specific features such as hashtags, user_mentions and URLs that appear in the retrieved tweets are also extracted and added to the list of features. For example, for the Canadian debate

about "Bill C–23, Fair Elections Act" the features "fraud", "#unfairelectionsact", "@PierrePoilievre" and the expanded URL of "http://fw.to/oO9okOb" are added, which represent respectively, a common term, a popular hashtag against the bill, the politician who introduced the bill, and a link to a news article explaining the bill. These newly found features are also added to the list of features. Therefore, the list of features can contain keyterms, hashtags, URLs, or user_mentions.

The discriminative power of our different types of features (terms, URLs, hashtags and user_mentions) is not necessarily the same. Thus, the non-zero values of $\omega_{i,j}$ are initially set according to the feature type. The overall intuition is that some features (e.g. hashtags) are more reliable indicators of the debate than other types of features (e.g. user_mentions), so we assign them different initial weights. The setting of these weights by feature type is further described in Section 3.4.2.

3.3.2 Active Tweet Retrieval

We propose four strategies for improving retrieval accuracy with user involvement. The goal is to select instances to be labeled that are most helpful in improving retrieval results while minimizing the number of labeling requests. The feedback resulting from a labeling request is important not only for that particular request, but also from what can be learned from it.

Strategy 1: Ambiguous Retrieval

Tweets are retrieved to the debate that has the highest similarity to them. However, multiple debates could have very similar highest scores. This scenario suggests a good opportunity for asking the user to clarify the ambiguity. The user feedback is useful not only to determine the correct debate for these similarly-scoring tweets, but also to modify the current list of automatically extracted discriminative features.

Let $sim(t_i, d_j) \approx sim(t_i, d_r)$ with d_j and d_r having the highest scores for t_i compared to other debates in D. Upon the presentation of t_i to the user, if she assigns t_i to d_j, we can find the specific features in $\mathbf{k}$ that contributed to $sim(t_i, d_r)$ and reduce their associated weights for debate d_r, i.e. reduce non-zero $\omega_{p,r}$ for any k_p in t_i. Similarly, an increase in the weights of the features of the winning debate is also applied. This reduction (or increase) is proportional to the overall number of tweets

associated with d_r for which feature k_p occurs in.

Any specific hashtag found in t_i represents a valuable piece of information (in Section 3.3.2 we discuss how specific hashtags are identified) that can be used for retrieving more relevant tweets from the set of non-retrieved tweets. In other words, when hashtag h_o that occurs in labeling request t_j is a specific hashtag, the retrieval method use this hashtag to retrieve more tweets for the same debate that the user assigns t_j with.

Strategy 2: Near-Duplicate Detection

We observed that in our Twitter dataset a large number of tweets are near-duplicates of each other (and they are not retweets). It is safe to assume that tweets that are near-duplicates should be assigned to the same debate. Therefore, we identify clusters of near-duplicate tweets, where a tweet is added to a cluster if it is a near-duplicate of all the tweets already in that cluster. Tweets from larger clusters have more potential as candidates for labeling requests, since labeling a single tweet from a cluster is sufficient to associate all its tweets with the same debate. Furthermore, the likelihood of the cluster belonging to a debate is an important criterion to avoid requesting the labeling of tweets in a non-relevant cluster. As a result, we rank clusters by their likelihood to belong to a debate and their cardinality, and use this rank to present to the user tweets from these near-duplicate clusters for labeling. We also take advantage of any specific hashtag identified in a labeled cluster, as described in the previous strategy.

Checking for near-duplicates naively has a quadratic time complexity, which given the typical scale of Twitter datasets, such a task becomes computationally prohibitive. Therefore, we apply Locality Sensitive Hashing [254], that allows near-duplicates to be found in linear time complexity. Intuitively, this method works by breaking down the text content into smaller pieces and applying a hash function to each piece. If multiple pieces of different documents are hashed to the same values, then there is a high probability that these documents are duplicates or near-duplicates. The specific approach adopted is similar to the one described by Soto et al. [256], which consists in taking word tri-grams of the tweet content, using min-hashing to reduce the feature set, and hashing signature bands to identify near-duplicate content.

Strategy 3: Leveraging Hashtags

Hashtags are possibly the most popular feature in Twitter. People use them to make their tweets easier to find and to engage in conversations with others. This third strategy relies on frequent specific hashtags to find tweets that were either retrieved to the wrong debates or failed to be retrieved. The sequence of steps for leveraging hashtag occurrences and user knowledge to improve retrieval results is illustrated in Fig. 3.2 and described next.

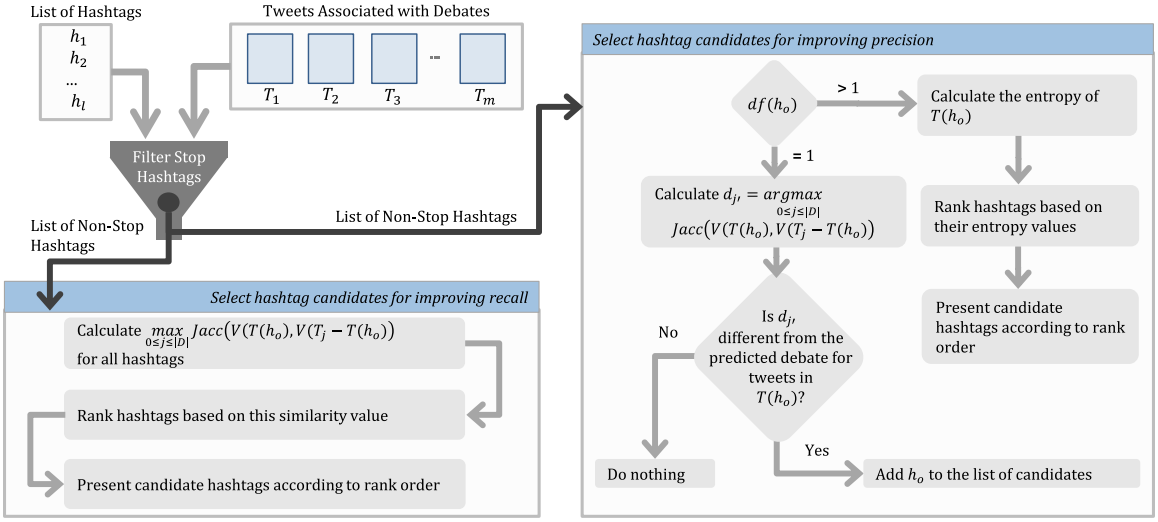


Figure 3.2: Hashtag selection strategy to improve retrieval precision and recall

Filtering Stop Hashtags Candidate hashtags for improving retrieval accuracy are those likely to be good indicators of a specific topic in the problem domain. For instance, while "#cdnpoli" is a hashtag often associated with posts related to Canadian politics in general, it cannot be taken as a discriminative feature in the context of our debate association problem, since it appears in tweets related to many different debates being held in the Canadian Parliament. We refer to these hashtags as "*stop hashtags*"—similar to stop words in natural language processing, they do not add any useful information for the topical categorization of the data.

Definition 4 *We define a virtual document as the concatenation in no specific order of the textual content of a set of tweets. We use the function $V(T')$ to indicate the virtual document obtained from the set of tweets T'.*

It is worth noting that stop hashtags are highly domain-specific. In our scenario, in order to identify a stop hashtag we must look at how many debates are likely to retrieve tweets, given that hashtag. The more debates a hashtag appears in, the more likely it should be considered as a stop hashtag. Therefore, we construct a virtual document for each debate by concatenating all its tweets as assigned by the retrieval algorithm, i.e. $V(T_j)$, $1 \leq j \leq |D|$. Then, each hashtag h_o is ranked in ascending order based on its debate frequency $df(h_o)$, i.e. the number of virtual documents that include hashtag h_o. The result is a sorted list of hashtags, with those on the top being more likely to be specific.

Improving precision using hashtags Our unsupervised method may retrieve content incorrectly, thus associating hashtags with the wrong debates, which in turn may lead to the retrieval of more irrelevant tweets. In order to use hashtags to improve retrieval precision we must first identify those non-stop hashtags that appear in tweets that have been incorrectly associated. Two scenarios are possible: either those tweets that include a non-stop hashtag have been retrieved to more than one debate, or all of them have been associated with a unique debate. The first scenario is likely to be a conflict situation for the system, as a specific hashtag has been retrieved to multiple debates. The second scenario would indicate that either all retrieved tweets containing that hashtag are correctly associated, or they are all incorrectly associated.

Definition 5 *Let $T(h_o)$ be the set of tweets that include a hashtag h_o and $F(T(h_o))$ the frequency distribution of debates associated with the tweets in $T(h_o)$ by our retrieval method. The normalized entropy for the distribution $F(T(h_o))$ is calculated as follows:*

$$\eta(F(T(h_o))) = \frac{\sum_{j=1}^{m} -F(T(h_o))(d_j)\, log(F(T(h_o))(d_j))}{log(|F(T(h_o))|)} \tag{3.2}$$

In the first scenario, h_o occurs in more than one debate, $df(h_o) > 1$. Assuming that h_o is not a stop hashtag, the value $df(h_o)$ is small. We first compute $F(T(h_o))$ and then its normalized entropy, $\eta(F(T(h_o)))$, using Eq. 3.2. After computing this entropy value for all the non-stop hashtags, they are ranked in decreasing order. This normalized entropy allows to identify uniform-like distributions in $F(T(h_o))$, which could be an indicator of incorrect associations as we assumed h_o to not be a stop

hashtag. Therefore, high-ranked hashtags are good candidates for involving the user to decide whether there is any issue with the retrieval of the tweets in $T(h_o)$ or not.

In the second scenario, the hashtag h_o occurs in a single debate, $df(h_o) = 1$. In this case, either all tweets in $T(h_o)$ have been correctly retrieved, or all of them have been assigned to the wrong debate. To determine which the true situation for the given hashtag h_o is, we first build a virtual document—referred to it as $V(T(h_o))$—by concatenating the textual content of all the tweets in $T(h_o)$. Then, we construct a virtual document for each debate by concatenating all the tweets previously associated with it except those in $T(h_o)$, i.e. $V(T_j - T(h_o))$. By calculating a similarity score between $V(T(h_o))$ and $V(T_j - T(h_o)), \forall j$, it is possible to identify the most similar debate to the tweets in $T(h_o)$. If the highest-scoring debate for the tweets in $T(h_o)$ is different from the debate retrieved by our algorithm there is a good likelihood that these tweets have been mistakenly associated. In this case we involve the user to address this inconsistency and decide which debate they should be associated with.

The similarity score between two virtual documents is computed from a vector profile built for each of them containing a list of words with the highest tf-idf values. Then, we compute the Jaccard similarity using binary weights for their corresponding vector profiles, i.e. $\text{Jacc}(V(T(h_o)), V(T_j - T(h_o)))$.

Improving recall using hashtags The unsupervised retrieval algorithm may fail to retrieve all relevant tweets mostly due to the vocabulary mismatch problem. To improve retrieval recall, we need to look for hashtags that have not been selected as discriminative features but are still "good indicators" of debates in D. A straightforward approach would be to select the most frequent non-stop hashtags occurring in the non-retrieved tweets (see Section 3.4.1 for a description of what is considered as our pool of non-retrieved tweets). However, these would not necessarily indicate similarity to any of the given debates. For instance, "#no2niki" is a frequent hashtag in tweets opposing a parliament member with reference to a motion on abortion. While it could be a discriminative feature to identify tweets about this topic, if "Abortion" is not among our selected debates, then this hashtag is not a good candidate for improving recall.

To identify non-retrieved hashtags that are indicators of debates in D, we first

follow the same previous approach to calculate the similarity between the virtual document $V(h_o)$ of the tweets that include a given hashtag h_o and all tweets retrieved as relevant to each debate in D. We consider the Jaccard similarity between non-stop hashtag h_o and debate d_j to be an indicator of the probability that h_o would be a discriminative feature for d_j. Therefore, the maximum similarity of h_o to different debates, i.e. max Jacc$(V(T(h_o)), V(T_j - T(h_o))), \forall d_j \in D$ presents the highest probability that h_o would be an indicator of one of the debates in D. In this way, a list of hashtags ranked in decreasing order of their associated maximum similarity value to the debates is built. Given that they are likely to be relevant to some debate in D, we follow the ranking to present the candidate hashtags in this list to the user and ask for feedback on their relevance.

Strategy 4: Leveraging Replies

As a social media platform, Twitter enables users to engage in conversations by replying to other users' posts. We consider this relational information between tweets as one of the selection strategies, following the hypothesis that replies to a tweet t_i are likely to be associated with the same debate as t_i.

We first trace back reply tweets to their *sources*, which are tweets that are not replies to any other tweet. We group together all the reply tweets that share the same *source*, including the *source* itself. The *reply chain* of a tweet t_i, i.e. $R(t_i)$, contains all tweets that have been grouped due to their reply relation.

Considering only the tweets in $R(t_i)$ that are already retrieved to debates in D, we calculate an entropy value using a similar approach to that described in Eq. 3.2. If all these tweets are associated with the same debate, then the entropy value will be equal to zero, while if they are split uniformly among all debates, the entropy value will be equal to one. Reply chains with a high entropy signal some inconsistency between the conversation topic addressed by the tweets and their retrieval, and consequently it is more likely that some of these reply tweets were indeed retrieved to the wrong debates. Therefore, tweets in high entropy reply chains are good candidates for user involvement. Their cardinality is also considered in selecting the candidate reply chains. Thus, reply chains are sorted based on their entropy value multiplied by a value proportional to their cardinality, i.e. $\eta(F(R(t_i)) \times log(|R(t_i)|)$. These reply

chains are presented to the user following this sorting. This strategy helps improve both precision and recall, as it allows to correct mistaken assignments and also to recover tweets that had not yet been retrieved.

It is not possible to calculate the entropy function if no tweets are retrieved in the reply chain of a source tweet t_i, i.e. $F(R(t_i)) = \varnothing$. This may be due to either a failure of the retrieval method, or these tweets are actually not related to any of the target debates. In order to determine whether these tweets are likely to be relevant, we select the largest reply chains for user inspection. In this case, any tweet labeled as relevant to one of the debates will contribute to improving retrieval recall. As in previous cases, any specific hashtag identified within a labeled tweet is leveraged as described in the first two strategies. Even if the user believes that the entire reply chain is not relevant to any debate, the method still benefits from the feedback by identifying specific hashtags and URLs that appear in these tweets and influencing the algorithm not to retrieve tweets containing these features in the future.

3.3.3 Interactive Visualizations

In order for the framework to be accessible and usable by users, the retrieval process and its embedded strategies have been integrated into a visual interface that includes multiple complementary interactive visualizations. The resulting tool affords a user-driven analysis that fosters a better understanding of the retrieval strategies and enables the system to incorporate user domain knowledge in learning the assignment strategies. Therefore, the visual interface of ATR-Vis has been designed to meet the following goals:

1. Provide an interface that incorporates and supports active retrieval strategies for an accurate and complete retrieval of tweets given a set of predefined debates. Such an interface should reduce user effort when handling labeling requests by presenting appropriate visual aids to support her task. To generate the desired impact, the tool should be understandable by non-data mining experts.

2. Enable user-driven exploration of the retrieved and non-retrieved tweets and allow her to modify the retrieval model as a result from this exploration, beyond the handling of the labeling requests (e.g. by updating the debate-characterizing

features).

ATR-Vis is a web-based application. The front end was built using D3.js [25] and Bootstrap [208], while the backend was written in Java and uses Apache Lucene [76] for text indexing and searching. Design choices of ATR-Vis were made based on the analytical tasks and the type of data to be visualized, and by following expressiveness and effectiveness principles [199]. Multiple coordinated visualizations are employed due to their helpfulness for exploring intricate data with diverse attributes [201].

The visual interface consists of two main views: *Assignment* and *More*. The *Assignment* view enables the retrieval method to obtain feedback from labeling requests as well as multiple secondary views for aiding the user in the manual association process and the analysis of the retrieved tweets. The *More* view was designed to accommodate the two strategies that make use of the structural information of the tweets: *leveraging replies* and *hashtags*. A demo video of ATR-Vis is available online[4].

Assignment View

The Assignment View is shown in Fig. 3.3. Each debate is uniquely associated with a color that is consistently preserved throughout the interface. The system presents the user a tweet labeling request selected either by the *Ambiguous retrieval* or the *Near-Duplicates* strategies, to be assigned to one of the predefined debates (panel *a* in the figure). To assist with the manual association of the current tweet, its non-stop words are shown with the color of the debate in which they occur more frequently.

Debates are presented as a vertical list (panel *c*), which follows the principle of grouping instances (tweets) assigned to the same category (debate) in a same spatial region [199]. Each debate shows its name and its similarity score with the current tweet in the labeling request. The similarity score is shown explicitly as a number, and also as a horizontal bar with length proportional to the score value, while always spanning to the panel width for the top-scoring debates. The reason for this design choice is that length is a preattentive attribute of visual perception and one of the most effective channels for encoding quantitative values [283].

Whenever the user hovers the mouse over a debate, a sample tweet associated to that debate is shown. She can also click on a debate to access all its assigned tweets

[4] `https://drive.google.com/file/d/0Byrh43zBaFKGMUxTS3A4YV9ubWM/view?usp=sharing`

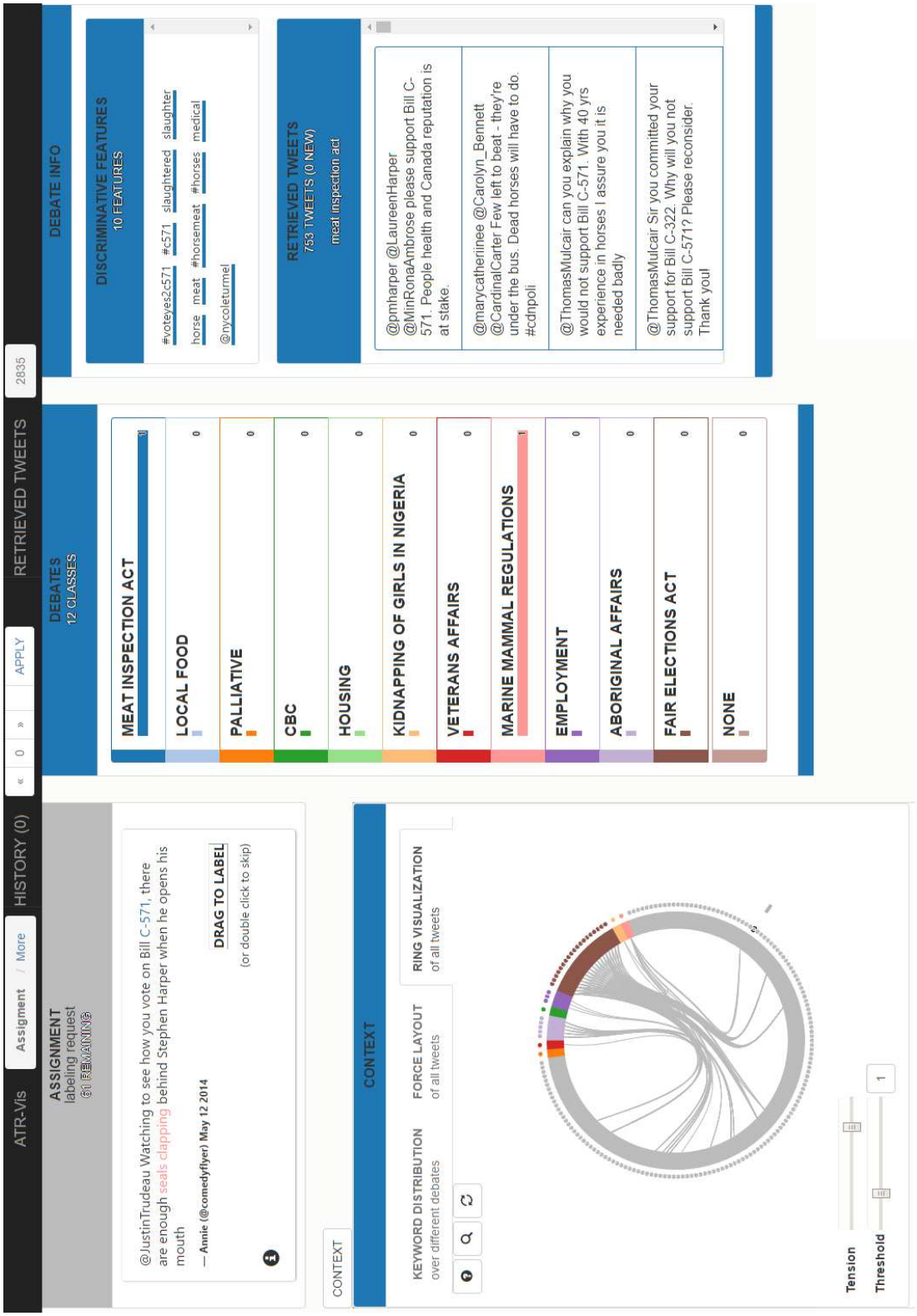


Figure 3.3: Assignment View: a set of visual aids to facilitate tweet retrieval. (a) Labeling request for a Twitter post. (b) Visualization of the labeling requests in a broader context. (c) List of debates of interest. (d) Discriminative features for the selected debate. (e) Tweets retrieved by the selected debate.

and discriminative features on a side panel. All background panels of the interface are colored according to the selected debate, except for the *Assignment* panel, which is colored with the debate that has the highest similarity score with its displayed tweet. The set of system-extracted discriminative features is shown as a sequence of terms (panel *d*), and the feature weight is displayed by hovering the mouse over a feature. If the user believes that a discriminative feature shown is not appropriate to this particular debate, she can modify it accordingly by dragging the feature to the proper debate.

The context panel (panel *b*) includes multiple visualizations to inform the labeling task. The *Keyword Distribution* tab allows observing the frequency distribution of any non-stop words occurring in the labeling request over the debates. Bar charts are a suitable choice, because they are good for analytical and accurate comparison of a value across multiple categories [145]. Two other visualizations place the labeling request in the global context of the currently retrieved and non-retrieved tweets. Given the potentially large number of tweets, they show only a sample, which includes tweets from the labeling requests and a stratified sample of those retrieved by each debate. Since we aim at showing relationships between the tweets, both visualizations rely on a graph metaphor, where the nodes represent tweets. An edge connects a pair of nodes if their similarity score is above a user-selectable threshold for one common debate. The two visualizations differ in how nodes are laid out: by the forces exerted by the connections in the *Force Layout*, or arranged in a circle in the *Ring Visualization*.

The Force Layout is one of the most common solutions to show networked data [199]. As a result of the force-directed algorithms simulating the effect of spring-like physical forces acting on the edges and repulsive forces on the nodes, the resulting layout tends to show spatial clusters of similar, or highly related nodes. Graphs of this type are commonly employed to represent different Twitter visualizations from followers graph [123] to retweet relationships between users [195] and hashtag networks [133]. Our Ring Visualization is similar to a chord diagram with same-width chords and edge bundling to minimize cluttering and reveal high-level edge patterns [109]. Its radial network layout is a good choice for showing relationships between different categories [145]. Chord diagrams have been used for visualizing different

aspects of Twitter data elsewhere [78, 222].

Furthermore, we followed the principle of attention management in ATR-Vis [282], ensuring that a visual cue with the results of a user selection is provided [199]. For instance, when double-clicking on any tweet in any of the views, we immediately attract the user's attention to the assignment panel, which shows the tweet. Both the Force Layout and the Ring Visualization are coordinated with the assignment panel as the labeling request is highlighted in the graph.

Other possible interactions in this view include: labeling the current request, flipping through the list of requests, selecting new posts for inspection, visualizing all user requests, getting the closest neighbors (most similar posts) to a post, and resampling the nodes currently visualized. There is also a bar at the top of the screen for keeping track of the sequence of actions, which also serves the purpose of visualizing the impact of each change.

More View

This view, presented in Fig. 3.4, allows interaction with two of the active retrieval strategies introduced to leverage the reply chains and the hashtags co-occurrence. The left-most panel (panel f) shows the reply-based conversation of a source tweet displayed as a tree layout. Hierarchical tree networks are good to represent the structure of conversations [145, 47, 213], where the approximate number of replies and the number of different branches can be assessed at a glance. The specific reply chain is selected according to the number of tweets involved and its potential to reveal conflicting retrievals, as explained in Subsection "Leveraging Replies". The user can explore the posts in the conversation, aided by the color indicating to which debate each one was retrieved, and decide whether an individual post or a whole subsequence of replies should be labeled. This view is especially useful in situations where there is topic drift.

The second view shows hashtags deemed as relevant for the user to inspect and provide feedback regarding their specificity to any of the given debates. The information is presented as a bipartite graph where one node depicts a selected hashtag and the other nodes depict the debates (panel g). Since linewidth is also a preattentive attribute of visual perception [283], edge width is used to indicate the similarity value

that measures the relatedness of the hashtag to a particular debate as explained in Subsection "Leveraging Hashtags". When the user hovers the mouse over the nodes of debates, the usage distribution of the selected hashtag among the retrieved tweets for each debate is also shown. This is complemented with a panel on the right (panel *h*) that lists all the tweets containing the focus hashtag and retrieved to a specific debate. Similarly to the previous strategy, the user can label individual posts or all posts containing the target hashtag.

3.4 Evaluation

In this section, we present results after evaluating ATR-Vis from multiple perspectives. We first present the datasets employed in these studies, followed by the parameter settings used in the strategies, and quantitative results of our retrieval strategies as compared with alternative retrieval strategies by means of a simulated user or oracle. We also describe three use cases that aim at describing typical scenarios of exploratory analysis with our framework. Finally, a user-oriented evaluation is described where three target users have interacted with the system in a pair analytics session [13].

3.4.1 Datasets

Experiments were conducted on two parliamentary datasets. The first one refers to the Canadian House of Commons during the period 12-16$^{\text{th}}$ May 2014. We selected the 11 debates that received most attention in the parliament during that week (measured in terms of their overall length of discussion), since these were more likely to generate an expressive number of opinions in social media. The second dataset refers to 5 mainstream debates being held in the Brazilian Federal Senate from 25$^{\text{th}}$ to 29$^{\text{th}}$ May 2015. The title of these debates are presented in Tables 3.4 and 3.6. Transcripts of the selected debates were extracted from the respective parliament websites [205, 154].

We used Twitter's streaming API [272] to collect tweets during the weeks of interest. Since it returns a minor fraction of the total volume of tweets at any given moment (roughly 1%), we must, to the maximum extent, restrict our search to Canadian (or Brazilian) political tweets in order to gather as many relevant tweets as possible without introducing many spurious ones. Furthermore, the collection procedure should

Figure 3.4: More View. (f) Exploring a conversation thread on Marine Mammal Regulations. (g) Exploring how the hashtag *#unfairelxnsact* is associated with different debates. (h) Enumeration of tweets containing a hashtag.

Table 3.2: Statistics of collected Canadian and Brazilian Twitter datasets. Columns from left to right represent: number of original tweets (without retweets), number of tweets containing at least one hashtag, number of tweets containing at least one URL, number of tweets containing at least one user_mention, and number of reply tweets.

Dataset	Tweets	Hashtags	URLs	User_mentions	Replies
Canadian	16297	7894	3835	15562	7783
Brazilian	9625	1364	2493	8418	3763

not be biased towards any specific keywords, or the resulting datasets would depend on the choice of keywords and will likely report these keywords as being relevant. Bearing this in mind and in agreement with recommendations by Miranda Filho et al. [74], we used as initial information the parliament members' Twitter accounts. Twitter user names for the Canadian Parliament members were obtained from the Politwitter website [221], while for the Brazilian case 80 Twitter user names, out of 81 senators, were identified manually. We collected tweets that were either posted by a parliament member, replied to a post by any of them, or that included one of their Twitter user names in its text. This resulted in datasets containing 16,297 and 9,625 original tweets (no retweets) for the Canadian and the Brazilian data, respectively. We chose to ignore retweets since they would be automatically retrieved to the same debate as their original tweets, distorting the retrieval results. The number of tweets containing at least one hashtag, URL or user_mention in each dataset is reported in Table 3.2. It also reports the number of tweets that are reply to another tweet as well.

Although our algorithm assumes that labeled data is not available—with the exception of the active retrieval strategies, where a user provides a few labels—we still need a subset of labeled tweets for evaluation purposes. Yet, sampling this subset is far from trivial. The most unbiased strategy would be to randomly select the instances to be labeled. Let us refer to a sample obtained with this approach as the sampling subset A. Therefore, for the Canadian data we manually labeled 1,000 randomly sampled tweets. However, we observed that most of the retrieved tweets (719 out of 1,000) are not related to any of the target debates. Adopting such a strategy would require us to label several thousands of tweets in order to gather sufficient

tweets for each target debate, which would evidently be infeasible.

Our alternative sampling strategy was then to randomly select a percentage of the tweets that are retrieved for each debate by our proposed algorithm. This subset suffices for us to estimate the precision of the proposed retrieval method, but in order to evaluate the recall we must also randomly sample from the non-retrieved pool of tweets. So, from this latter group we randomly sampled as many tweets as the number sampled for the most popular debate. Overall, we manually labeled 2,634 additional tweets for the Canadian data (comprising a sampling subset B), in addition to the 1,000 randomly sampled tweets (subset A). In the case of the Brazilian data, for a good compromise between benefit and labeling effort we only applied the second strategy (sampling B), obtaining 1,064 labeled tweets. Our code and the resulting datasets, including the labeled subsets, are publicly available[5].

3.4.2 Parameter Setting

To avoid retrieving tweets that are only loosely related to a debate, we assign tweets to the debate with the highest similarity score if that score is above a given threshold, which in our experiments we set to 1.0. As discussed in Section 3.3.1, the values in matrix Ω indicate the relevance of the extracted features to the different debates, and hence their contribution to the similarity scores of tweet-debate pairs. Based on an initial appreciation of feature importance, these weights have been initialized to 1.5 for hashtags, 1.0 for keyterms and URLs, and 0.5 for user_mentions. Weights are adapted when employing the *ambiguous retrieval* strategy, as described in Section 3.3.2.

The initial number of features is $|D|\times\kappa$ keyterms, $|D|\times\mu$ user_mentions and URLs, and $|D| \times \tau$ hashtags, where κ, μ and τ were set to 5, 2 and 1, respectively. The active retrieval component requires setting a single parameter, namely the minimum debate frequency for filtering stop hashtags. As this value should be proportional to the number of debates, in our experiments we filtered out hashtags that appear in over a fourth of the debates, i.e. $df(h_o) > (|D|/4)$. The sensitivity to parameter settings is discussed in Section 3.4.6. We adopted the same parameter settings on both datasets.

[5] `https://drive.google.com/drive/folders/0Byrh43zBaFKGa2xpWGFVamVJWGM?usp=sharing`

3.4.3 Retrieval results

We consider multiple evaluation metrics in reporting our results, namely *accuracy*, given by the ratio of correctly retrieved tweets to the total number of labeled tweets, and *macro-precision* and *macro-recall*, due to the strong imbalance in the distribution of tweets in the debates. To calculate macro-averages, we consider the same weight for each class, while for accuracy, individual instances contribute to the score with the same weight. Furthermore, we consider two additional metrics that take into account the ranking or scores of the retrieved instances. One is *Mean Average Precision* (MAP), which indicates the average precision of the results across different levels of recall. The other metric is *R-precision*, which measures the precision of the top R retrieved documents, where R is the number of known relevant documents [180]. We also report *precision*, *recall*, *R-precision* and *MAP* for each debate. Appendix G show how these evaluation metrics are calculated.

As per our initial assumption, we found out during labeling that most tweets in our datasets are single-labeled. For those few multi-labeled tweets we consider their retrieval to be correct if they are associated with any of their multiple debate labels. Note that although we can only report the results for the labeled tweets, we perform the retrieval steps and selection strategies considering all tweets in T, all of them having the same probability of being selected for labeling requests regardless of their labeled status.

A summary of the performance of the unsupervised retrieval method and the various active retrieval strategies on the Canadian dataset is presented in Table 3.3, which also includes the number of labeling requests needed by each retrieval approach. For the unsupervised method, we report two separate results. One refers to the method after its first iteration, i.e. considering only the keyterms extracted from debates as discriminative features, while the other refers to applying the full unsupervised method, i.e. considering pseudo-relevance feedback from the tweets to expand the list of features. We can see from Table 3.3 that the pseudo-relevance allows retrieving many more relevant tweets while retaining a comparable precision.

To examine the impact of the proposed selection strategies on improving retrieval accuracy, we first applied each strategy separately and then calculated the number of tweets that were correctly retrieved as a result of its application. When using

Table 3.3: Accuracy, macro-precision, macro-recall, R-precision and MAP for the unsupervised retrieval method, a random active retrieval strategy, ReQ-ReC, and the ATR-Vis' selection strategies using the Canadian dataset. *The number of labeling requests (#Req.) reported for ReQ-ReC is an average over all debates.

Retrieval method	Accuracy	Macro-Pr	Macro-Re	R-precision	MAP	#Req.
Unsupervised (1^{st} iteration)	0.61	0.74	0.55	0.67	0.70	0
Unsupervised	0.80	0.75	0.68	0.70	0.71	0
Random active retrieval	0.81	0.76	0.70	0.71	0.73	100
ReQ-ReC	0.29	0.26	0.70	0.66	0.64	116*
Ambiguous retrieval (1)	0.83	0.80	0.75	0.75	0.75	15
(1) + Near-Duplicates (2)	0.84	0.81	0.76	0.76	0.76	24
(1) + (2) + Hashtags (3)	0.89	0.82	0.81	0.79	0.80	60
(1) + (2) + (3) + Replies	0.92	0.83	0.86	0.82	0.84	100

the hashtag-based strategy, for each non-stop hashtag h_o we simulate the user by randomly selecting three labeled tweets from $T(h_o)$. If the three tweets are labeled with the same debate, then all tweets including this hashtag will be considered as belonging to that debate. A similar approach was adopted when simulating the user in the reply-based strategy. The results for each strategy, on 100 labeling requests (divided into blocks of 10 requests), are illustrated in Fig. 3.5. The line representing the ambiguous retrieval strategy is shorter because only 20 tweets from our test set have nearly equal scores to their most similar debates. We also report the results of applying all the selection strategies in sequence (in the same order as described and reported in Table 3.3). The results highlight that combining these four strategies for retrieval outperforms the independent application of any of them. We also performed McNemar test to determine whether the improvement in the retrieval performance is statistically significant. We observed that all of our selection strategies significantly improved the accuracy of the retrieval compared to the previous steps, the unsupervised method, and both active baselines (with $p < 0.001$).

Evaluation metrics of the retrieval methods relative to each Canadian debate are presented in Table 3.4. We can see that for most debates the proposed active retrieval method effectively improves both precision and recall, while for some debates it only improves one of these measures while the other drops, or improves much less so. For instance, there is an increase in recall and in precision respectively, for the debates "Fair Elections Act" and "Marine Mammal Regulations". Moreover, the extent of

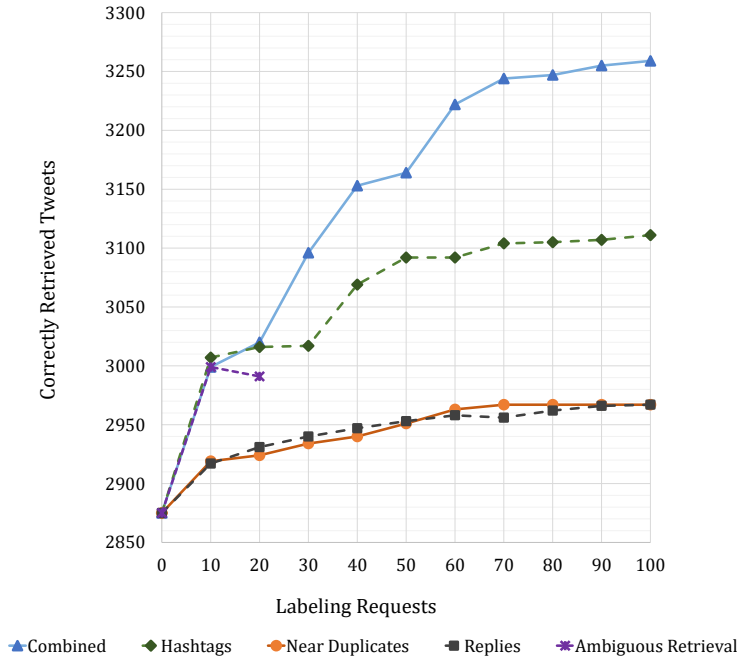


Figure 3.5: Canadian dataset: for each selection strategy, the number of correctly retrieved tweets. Blue triangles show the results of applying the four strategies in sequence.

the improvement varies: it is expected that on debates for which the unsupervised method already performs extremely well, as it is the case of "Meat Inspection Act", the improvement introduced by the active retrieval strategies is not as significant as in those for which the unsupervised retrieval is not as effective, as in the case of "Marine Mammal Regulations".

To further evaluate the effectiveness of the selection strategies, we compare the retrieval results with a random-based selection strategy, which serves as a baseline for our approach. 100 instances are randomly selected and their labels requested from the user. Similarly to our active retrieval strategies, labeled tweets are used to expand the list of discriminative keyterms, which in turn results in more tweets being retrieved. Likewise, we identify specific hashtags that occur in these user-labeled tweets and add them as discriminative features to their corresponding debates. The evaluation measures for this experiment are also included in Table 3.3, which shows averages over 10 runs with randomly selected labeling requests.

Table 3.4: Results obtained with the unsupervised retrieval method and the ATR method, for each debate in the Canadian dataset. Debate abbreviations stand for: Fair Elections Act (FEA), Meat Inspection Act (MIA), Employment (EMP), Aboriginal Affairs (AAF), Veteran Affairs (VAF), Kidnapping of Girls in Nigeria (KGN), Canada Broadcasting Corporation (CBC), Marine Mammals Regulations (MMR), Housing (HOU), Local Food (LFO), Palliative and End of Life (PAL).

		FEA	MIA	EMP	AAF	VAF	KGN	CBC	MMR	HOU	LFO	PAL
	#Labeled tweets	670	536	365	394	125	122	101	73	22	8	9
Precision	Unsupervised	**0.98**	**0.98**	**0.98**	**0.9**	**0.99**	**0.92**	**0.79**	0.67	0.89	0.09	0.18
	ReQ-ReC	0.47	0.54	0.35	0.31	0.12	0.14	0.24	0.11	0.09	0.25	0.24
	ATR-Vis	0.97	**0.98**	0.89	**0.9**	0.97	0.89	0.78	**0.88**	**0.91**	**0.35**	**0.67**
Recall	Unsupervised	0.65	0.96	0.63	0.8	0.56	0.81	**0.97**	0.51	0.36	0.37	0.67
	ReQ-ReC	0.64	0.84	0.8	0.68	**0.96**	0.85	0.55	**0.8**	0.9	0.22	0.45
	ATR-Vis	**0.94**	**0.99**	**0.92**	**0.81**	0.61	**0.92**	**0.97**	0.78	**0.95**	**0.75**	**0.89**
R-Prec	Unsupervised	0.87	0.96	0.87	**0.81**	0.61	0.83	0.85	0.15	0.73	0.37	0.67
	ReQ-ReC	0.59	0.82	0.72	0.62	**0.91**	0.85	0.55	0.66	0.87	0.22	0.45
	ATR-Vis	**0.94**	**0.98**	**0.89**	**0.81**	0.64	**0.89**	**0.86**	**0.79**	**0.91**	**0.5**	**0.78**
MAP	Unsupervised	0.91	0.97	0.87	**0.81**	0.77	0.88	**0.88**	0.12	0.78	0.19	0.67
	ReQ-ReC	0.49	0.79	0.73	0.59	**0.9**	0.83	0.48	0.71	0.87	0.22	0.45
	ATR-Vis	**0.95**	**0.99**	**0.91**	0.78	0.76	**0.91**	0.85	**0.77**	**0.95**	**0.59**	**0.85**

We also compare our approach with ReQ-ReC [155], a state-of-the-art active retrieval method. ReQ-ReC executes two iterative loops, where the outer-loop is responsible for improving recall by forming new queries and retrieving additional relevant tweets, while the inner-loop's job is to maximize the precision of the retrieved tweets. The labeling requests are selected considering the uncertainty of an SVM classifier, which is trained as part of the method's inner-loop. At each inner-loop iteration, 10 tweets with the minimum distance to the decision boundary, i.e. 5 from each side, are selected as the labeling requests. Once these requests are labeled by the user, the classifier is retrained and applied to the corpus of retrieved tweets. The inner loop ends when the classifier performance converges. Then, a new query based on the retrieved tweets is formed in the outer loop and used to recover additional relevant tweets.

ReQ-ReC's authors compared variations of their method regarding expanding the query based on retrieved instances. Their "Active" method, which uses Rocchio's method for query expansion [234], outperforms other variations for query expansion. Since ReQ-ReC works under the assumption of having one single query at a time, it has been applied separately to each debate. The results of applying the slightly modified version of the original ReQ-ReC (Active method) to the Canadian political dataset are shown in Tables 3.3 and 3.4. The number of labeling requests (#Requests) for ReQ-ReC in Table 3.3 indicates the average number over all debates.

Table 3.5: Accuracy, macro-precision, macro-recall, R-precision and MAP for the unsupervised retrieval method, and the ATR-Vis' selection strategies using the Brazilian dataset.

Retrieval method	Accuracy	Macro-Pr	Macro-Re	R-precision	MAP	#Requests
Unsupervised (1^{st} iteration)	0.74	0.74	0.73	0.70	0.73	0
Unsupervised	0.71	0.72	0.77	0.71	0.66	0
Ambiguous retrieval (1)	0.73	0.73	0.82	0.73	0.67	3
(1) + Near-Duplicates (2)	0.78	0.79	0.82	0.74	0.69	24
(1) + (2) + Hashtags (3)	0.80	0.82	0.88	0.78	0.8	60
(1) + (2) + (3) + Replies	0.77	0.79	0.9	0.78	0.75	90

Results show that our method, which uses specific Twitter features, outperformed ReQ-ReC on all the evaluation metrics. Moreover, we observe that Macro-Precision for ReQ-ReC is much lower than R-Precision and MAP. This may happen since the system does a good job of finding relevant tweets in the first outer iterations and ranking them at the top of the retrieved tweets. However, most of the lower-ranked tweets retrieved may not be highly relevant to the debate, and since they are considered to formulate new queries, it leads to a deterioration of the overall retrieval precision.

We partly replicated the previous experiments considering the Brazilian parliamentary dataset. A comparison of the retrieval results before and after the application of our retrieval strategies is presented in an aggregated manner and segregated by debates in Tables 3.5 and 3.6, respectively. The results indicate that the pseudo-relevance feedback of the unsupervised approach increase the retrieval recall at the expense of a drop in the precision. This is somewhat expected as adding new features to the debates may introduce spurious tweets. However, the active learning strategies seem to identify the incorrectly retrieved instances as the retrieval precision surpasses the initial values while also succeeding in finding new relevant tweets that previously failed to be retrieved.

The results broken down by debates shed some light on the reasons for the retrieval performance. Despite being handled as separate bills at the Brazilian Senate, debates on the topics of social security changes and workers' rights reforms are tightly related. Therefore, it is only after some user feedback that false discriminative keyterms are identified. Better retrieval rates are attained on all debates after simulating user feedback, as depicted in Table 3.6. It is important to note that in all tables, "#Labeled

tweets" refers to the number of tweets that are manually labeled for each debate for the purpose of evaluating the results, while "#Requests" indicates the total number of labeling requests asked from the information source.

Table 3.6: Results obtained with the unsupervised retrieval method and the ATR-Vis' selection strategies, for each debate in the Brazilian dataset. Debate abbreviations stand for: Social Security (PRE), Workers' Rights (TRA), Political Reform (REF), Fiscal Adjustments (AJU) and Brazilian Development Bank (BNDES).

		PRE	TRA	REF	AJU	BNDES
	#Labeled tweets	89	116	175	121	129
Precision	Unsupervised	0.42	0.78	0.66	**0.9**	**0.82**
	ATR-Vis	**0.67**	**0.87**	**0.67**	**0.9**	0.81
Recall	Unsupervised	0.67	0.69	0.92	**0.83**	0.75
	ATR-Vis	**0.95**	**0.85**	**0.95**	0.81	**0.93**
R-Prec	Unsupervised	0.47	0.73	0.66	0.83	0.83
	ATR-Vis	**0.56**	**0.82**	**0.73**	**0.84**	**0.94**
MAP	Unsupervised	0.29	0.58	0.75	**0.81**	0.84
	ATR-Vis	**0.56**	**0.74**	**0.82**	0.74	**0.92**

3.4.4 Use Cases

We discuss and compare the application of ATR-Vis to three use cases. We first discuss use cases related to the Canadian parliamentary debates and Brazilian Federal Senate debates. In addition, to showcase the suitability of the tool to other domains, another use case featuring major international news during the period 15-27$^{\text{th}}$ July 2016 is also presented.

Canadian Parliamentary Dataset

We now describe how an analyst could use the ATR-Vis framework to retrieve relevant tweets about the Canadian parliamentary debates. The user is first presented the *Assignment* view, shown in Fig. 3.3. She may start exploring the list of tweets retrieved for each debate and the list of discriminative features, in case some of the retrieved tweets are unexpectedly associated with a mistaken debate. Assuming no major issue is identified in the retrieval, the user may focus on the labeling request presented in

the assignment panel (panel *a*), as addressing the request is likely to improve the retrieval accuracy considerably. In this case the tweet requested is "*@justintrudeau watching to see how you vote on bill c-571, there are enough seals clapping behind stephen harper when he opens his mouth*", and we can anticipate why the method has found evidence it could belong to the debate on Marine Mammal Regulations, which addresses the regulation of seal hunting in the arctic, and to the debate on "Meat Inspection Act", debated as bill c-571 (notice both debates are shown with a long horizontal bar, unlike the others in the debates list). Its manual labeling to the correct class ("Meat Inspection Act") will contribute to strengthening the evidence that keyterm "c-571" is a highly discriminative feature for this particular debate.

The Ring Visualization, shown in Fig. 3.6, allows the identification of connections among tweets retrieved by different debates, which is an indication of potentially conflictive retrieval. Let us assume that the user is interested in finding more tweets associated with the debate on "Palliative and End of Life" (shown in orange), which seems to be under-represented in the current retrieval. After exploring the connections between tweets related to Palliative and unretrieved tweets (shown in gray), she finds out that some tweets mention the apparent mentor of the Palliative Care bill (@stevenjfletcher), albeit without using any of its typical keyterms. This interaction could lead to its incorporation into the features characterizing this particular debate, and therefore contribute to increasing the retrieval recall.

The *More* view enables taking advantage of the tweets' structural characteristics to improve the retrieval process (Fig. 3.4). On the left hand side interesting reply chains according to the criterion described in Subsection "Leveraging Replies" are presented for user inspection using a space-filling tree layout. The one shown refers to a conversation around the topic of seal hunting regulation. Twelve of these tweets have been correctly retrieved to the debate "Marine Mammal Regulations" as indicated by their associated color. However, three tweets were incorrectly retrieved to the debate "Meat Inspection Act", whose inspection and correction represents an opportunity to improve retrieval precision for both classes. The user can notice that other slightly related topics are mentioned in some of the tweets, such as Canadian cultural aspects related to seal hunting. These tweets were not retrieved because they use a different vocabulary than the typical topics mentioned in the debate. Incorporating them

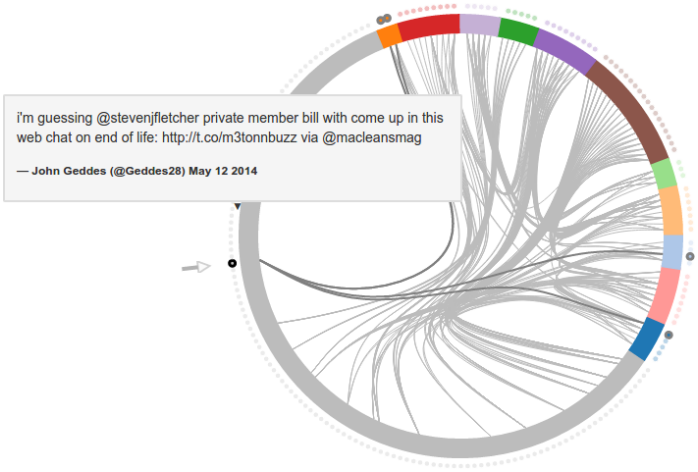


Figure 3.6: Ring visualization: identifying weak connections between unretrieved tweets and under-represented debates. Colors indicate different debates and two tweets are connected if their similarity score is above a user-selectable threshold for one common debate.

increases the recall and enriches the vocabulary associated with this debate. Also some "branches" of this conversation turn into an exchange of aggressive posts not at all discussing debates or political differences. The user can anticipate the nodes where this situation is likely to happen by inspecting branches with several non-retrieved (gray) nodes and refrain from labeling this branch.

In the middle panel of Fig. 3.4, specific hastags can be leveraged by supervising or correcting the most likely retrieval of the tweets that include them. The first nine hashtags shown in the figure are specific to the debates "Fair elections act" and "Employment", with the exception of "#jobs" which also appear in tweets related to other debates. The user can further inspect the tweets making use of each hashtag (on the right panel) and the bipartite graph depicting hashtag-debate connections for a selected hashtag. She may decide to label the hashtag as a discriminative keyword of a particular debate, with the additional option of automatically retrieving all the tweets including them to that debate. At any time the user can submit the modifications and see the impact of her interactions in the overall retrieval process.

If our user were a political journalist, there are several interesting findings that she could discover from the interaction with the tool. For example, the debate "Fair Elections Act" attracted by far the most attention in social media. By skimming over some retrieved tweets in this class, it can be noted that major concerns were raised by

Twitter users regarding the possibility of using the bill in an anti-democratic spirit by the government in power. In addition, by inspecting the discriminative features for this debate, a list of the URLs, which includes several news articles, can be identified. This can help find the most influential articles on Twitter.

Brazilian Federal Senate Dataset

As opposed to the previous use case, in this case we assume that the user starts interacting with the system from scratch. Therefore, at the beginning several features learnt from the debate transcripts are not completely discriminative. For instance, in the debate about "Social Security" (*Previdência Social*) the term *veto* is used extensively, while curiously other senators did not use it at all while discussing other debates that week. As a result, some of the initial labeling requests prompt the user to provide feedback when tweets talk about a veto for other bills (Fig. 3.7-a). The user can indicate the correct debate for these cases, which leads to a decrease in the weight of the term *veto* for the "Social Security" debate, or directly removing keyword *veto* from its discriminative features.

As indicated by the previous experiments, leveraging hashtags is the most effort-efficient way of providing feedback as it is likely to affect a considerable number of retrieved tweets. Due to alleged corruption cases in Brazil, senators and the general public requested internal investigations through a CPI (Portuguese acronym for "Parliamentary Investigation Committee"). Therefore, the system recognized frequently used hashtags like #cpibndes or #cpidacbf (Fig. 3.7-b). The first one is strictly related to the BNDES debate, whereas the second one is about investigating the Brazilian Confederation of Football. After submitting the appropriate feedback (jointly with the feedback on other hashtags), the user can observe a 70% increase in the number of tweets retrieved.

After a first batch of user feedback, features indeed become more specific, and the following labeling requests presented to the user are ambiguous even for humans. These difficult cases can also be identified from the graph visualizations. In the force-based graph, bridges and clusters of nodes with mixed colors are likely to indicate posts exhibiting some ambiguity regarding the features learned for each debate, as illustrated in Fig. 3.8-a with the debates "Fiscal adjustment" (*Ajuste Fiscal*) and

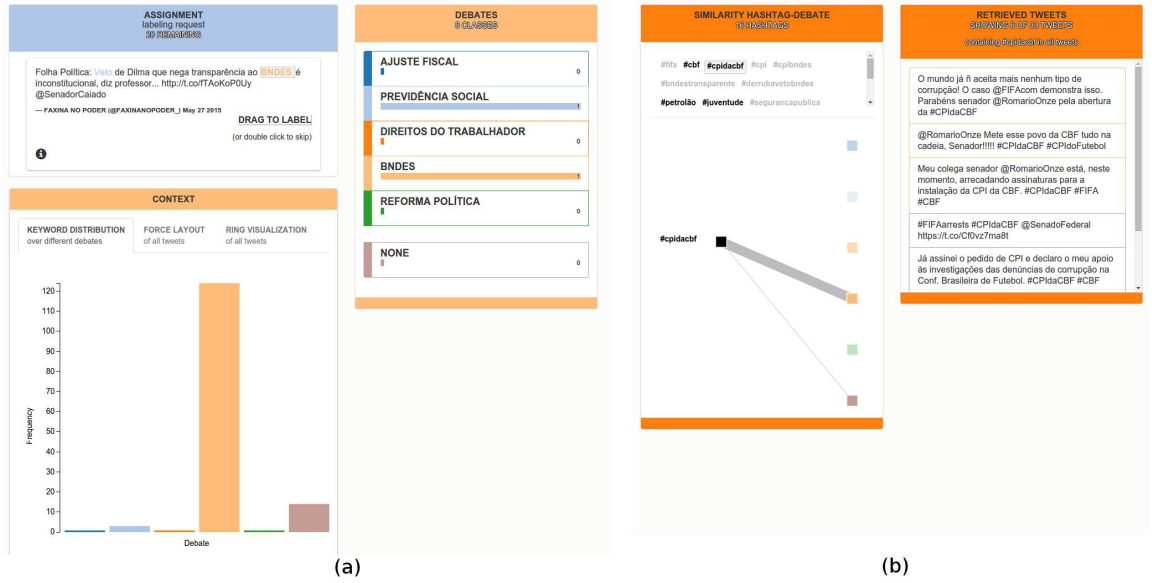


Figure 3.7: Beginning of the user involvement considering the Brazilian parliamentary dataset. (a) Ambiguous retrieval. The word "*veto*" is mistakenly associated to the debate "Social Security" (*Previdência Social*), and hence in conflict with the correct debate BNDES (b) Leveraging hashtags. The hashtags "#cpidacbf" is mistakenly associated to "BNDES" due to a transitivity with the hashtag "#CPI" which is also frequent in posts related with the debate "BNDES".

"Workers' Rights" (*Direitos do Trabalhador*). Similarly, by interacting with different link thresholds in the Ring Visualization, it is possible to identify additional non-retrieved tweets that are related to a specific debate. A tweet weakly connected to the debate "Fiscal adjustment" because it includes the word "adjustment" is illustrated in Fig. 3.8-b. There are also replies to these senators' posts, which potentially lead to the identification of additional relevant tweets.

Recent News Stories

In order to assess ATR-Vis outside the political domain and parliamentary debates, we considered news stories that received major attention from the media during the period 15-27 July 2016 as our set of topics, namely: "Terrorist Attack in Nice", "Brexit", "Colombia's Government and FARC", "Dallas Shooting", "Israeli-Palestinian Conflict", "Killing of Afro-Americans", "Orlando Nightclub Shooting", "Refugee Crisis", "Rio 2016 Olympics", "Turkey Attempted Coup", "US Presidential Campaign". We

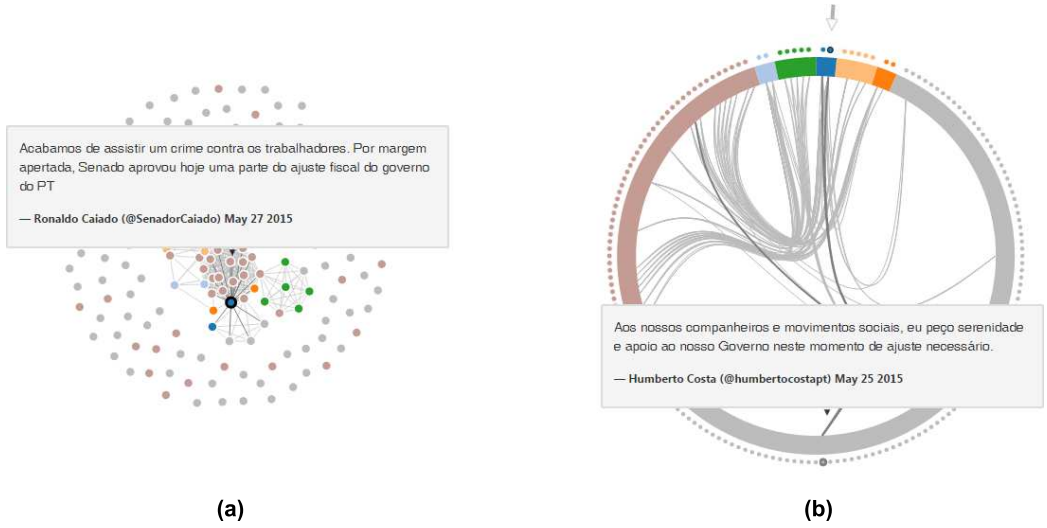


Figure 3.8: After a batch of user feedback using the Brazilian parliamentary dataset. (a) The tweet translates as "*We just witnessed a crime against workers. By a tight difference, the Senate approved today part of the fiscal adjustment proposed by the PT government*" and could be connected to the debates "Fiscal Adjustment" and "Workers' Rights" (b) The highlighted tweet, which translates as "*To our fellows and social institutions, I ask for your patience and support to our government during this time of necessary adjustments*", signals the association to the "Fiscal Adjustment" debate due to the presence of the word "adjustment".

considered news articles from CNN and Fox News related to each story to extract keywords and set our initial queries. The tweets collected during this period accounted for 9,277,751 after retweets and non-English posts were discarded. We have made this dataset also publicly available.[6]

We now describe how a user could interact with ATR-Vis to retrieve relevant tweets to these stories and even learn more about them. Due to several terrorist attacks and events of a violent nature that happened during the period we collected our dataset, the selected news stories have a high degree of similarity to each other, which makes the automatic retrieval of tweets a difficult task. The user can perceive this similarity between the stories through different visual components of ATR-Vis. The user may start with the Assignment View and the first suggested labeling request. As shown in Fig. 3.9 the labeling request is *"BoingBoing: RT AkyolinEnglish: 17 Turkish police officers killed in Ankara - by the junta-would-be. Horrible. It seems the coup wont be sub..."*. While "Turkey Attempted Coup" is the correct association for this tweet, the system also finds "Dallas Shooting" as a potential candidate, due to the presences of terms: "police" "officers" and "killed". After the user assigns the tweet to the correct event, the retrieval system also learns from this interaction to dampen the ambiguous terms for "Dallas Shooting" as they may also appear in other stories.

Then the user may want to focus on the context panel, where she can explore the connections between tweets based on their similarities to the stories. Fig. 3.10 shows that "Dallas Shooting" (in orange) and "Killing of Afro-Americans" (in pink) are strongly related stories. For instance, the inspected tweet: *"Watch: Baton Rouge: Timeline of Shooting: A look at the attack that left three police officers dead. https://t.co/UjqfSO8RXh"* contains elements from both stories as there is a mention to killed officers, and to Baton Rouge, the city where an Afro-American was killed. ATR-Vis is able to find the connections between these stories, and also find intermediate stories (appearing as a hub between two clusters) that may not belong to any of the stories of interest.

Let us assume the user is interested in finding more tweets relevant to the story "US Presidential Campaign" (shown in red), which seems to be under-represented.

[6] `https://drive.google.com/drive/folders/0Byrh43zBaFKGa2xpWGFVamVJWGM?usp=sharing`

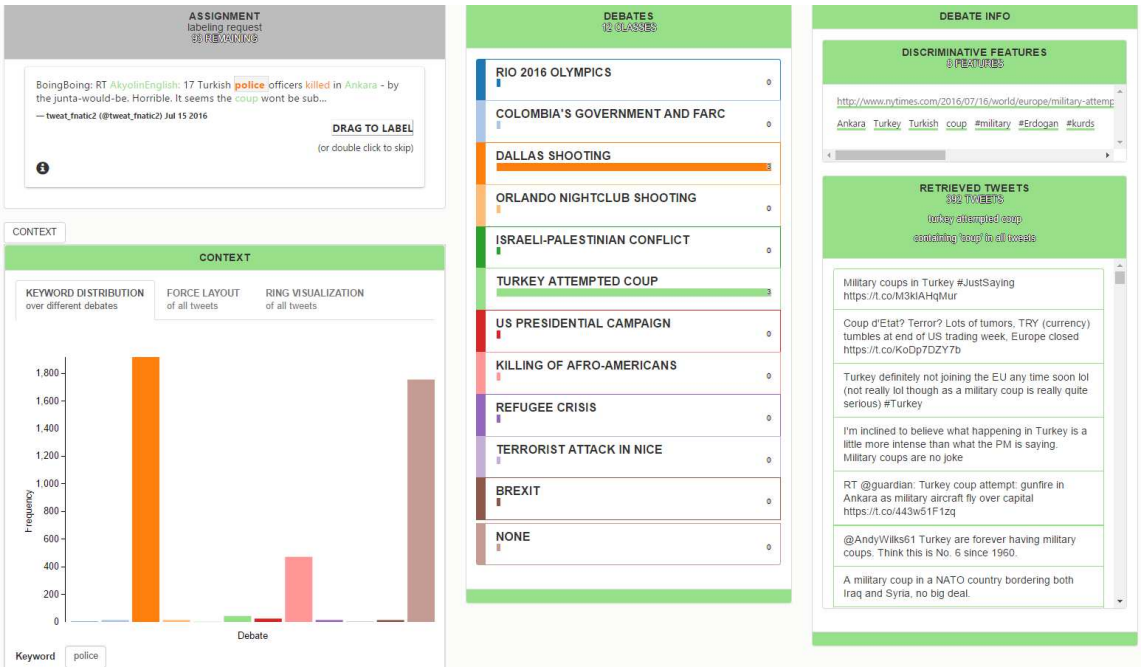


Figure 3.9: The Assignment View showing the labeling request, discriminative features, and keyword distributions for tweets on emerging news stories.

Figure 3.10: The Force Layout shows the tight connection between the story of "Dallas Shooting" (in orange) and "Killing of Afro-Americans" (in pink).

Exploring the connections between tweets related to this story and non-retrieved tweets (shown in gray), allows the user to find some relevant tweets, which in turn refines its discriminative features and improves its retrieval accuracy (recall and precision). For instance, Fig. 3.11 shows the tweet *"LA Times suggests MILITARY COUP when Trump wins Presidency https://t.co/oiRGbunXj3"*, which also has connections to tweets associated with the story "Turkey Attempted Coup". Since this tweet contains the terms "military" and "coup", ATR-Vis could not make a confident decision about its assignment. Associating this tweet with "US Presidential Campaign", results in adding the expanded URL of the tweet to the list of discriminative features for this story, which will be in turn used to retrieve more relevant tweets.

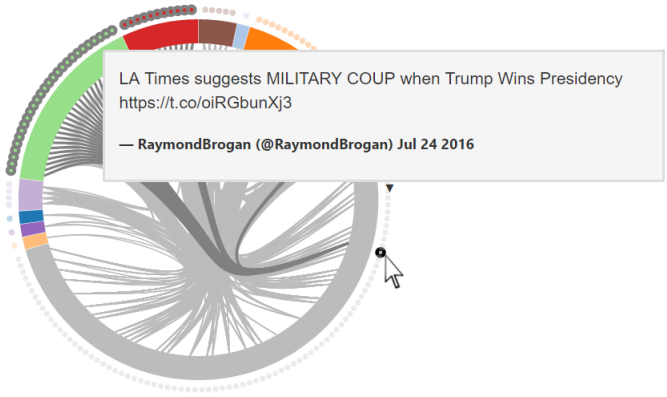


Figure 3.11: ATR-Vis can be used to improve the recall of a story. This image shows the connection of a non-retrieved with two of our stories, which the user can inspect to decide on its correct retrieval.

The last interaction is with the Similarity Hashtag-Debate. The identification of hashtags that are good indicators of our selected news stories and assigning them to the proper story can be an important contribution to improve recall and precision of the retrieved tweets. After reviewing different hashtags and the tweets containing them, the user can easily assign hashtags like "#CharlesKinsey", "#TurkeyCoupAttempt" and "#NiceFrance" to their corresponding story. In addition, there are other less obvious hashtags, like "#BlueLivesMatter", which after some inspection can be understood as a response to the hashtag "#BlackLivesmatter", in support for the officers killed in the "Dallas Shooting".

3.4.5 ATR-Vis Pair Analytics Evaluation

We already discussed the effects of our proposed strategies on improving the performance of the tweet retrieval by considering an oracle in Section 3.4.3. To investigate whether the proposed visualization is successful in supporting the retrieval process and helping the user better understand the data and make decisions, we performed an evaluation of the system with three domain expert users by means of a pair analytics process [13], which is carried out with one Subject Matter Expert (SME) and one Visual Analytics Expert (VAE).

We performed three pair analytics sessions, each involving one SME. Our first SME is a journalist and expert in online news and multimedia. As part of her daily work, she searches in Twitter to find interesting topics for stories and to identify active influential users to follow, study and interview with. She is also a university professor teaching journalism in the context of modern digital platforms. The second SME is a university professor in Sociology and Criminology, whose recent research interests include the portrayal of crime and racism in social media. Our last SME is a student taking a multidisciplinary degree in Criminology and Computer Science, and a research collaborator of the second SME. The latter two users are interested in understanding the discourse related to crimes in social media. In their opinion, analyzing social media is very important for social science researchers as people express their opinions more freely and more honestly than they would in answering questionnaires. We refer to the first, second, and third SMEs as SME1, SME2 and SME3, respectively.

Inspired by the work of Lam et al. [152], in this evaluation we followed the guidelines of two of their seven evaluation scenarios. The first scenario is "Visual Data Analysis and Reasoning" (VDAR), which discusses the evaluation of tools to support analytical tasks. Since these are typically complex and context sensitive, these evaluations are usually case studies with realistic tasks and domain experts. Questions for this scenario address ways in which the tool can help users to find the information they are seeking, form hypothesis and make decisions. The second scenario is "User Experience" (UE), which aims at evaluating people's opinions and their personal experiences about a tool, to what extent it was successful in assisting them to complete the tasks in their minds, and their suggestions for improvement. Questions for this

scenario address the user appreciation of the system, whether they would consider using ATR-Vis in their work/research, and their suggestions for improvements.

Before the evaluation session, we asked the SMEs to quickly familiarize themselves with the debates of our Canadian parliamentary dataset. They were given links to news articles and Wikipedia pages of the corresponding bills. We started the evaluation session by asking background questions, such as whether they need to search/analyze Twitter data in their profession, what information they look for and how they obtain this information; e.g. whether they use any external tools and whether they feel their information needs are satisfied. Then, we overviewed our dataset and explained the motivation of our system. We followed this by showcasing ATR-Vis along with its main interactive features. Then, the SMEs, assisted by the VAE, conducted the retrieval of tweets by interacting with different features of ATR-Vis. SMEs were encouraged to provide feedback and to review the effects of their interactions on the assignment of tweets, hashtags, and discriminative features to the debates. At the end of the session, they answered questions about ways ATR-Vis can be used to meet their information needs, comments for improving the system, and their general evaluation of the system.

Before Pair analytics with ATR-Vis: SMEs' Background

As part of her daily work, in order to find relevant information or stories about a topic, SME1 tends to search for an active Twitter account or a hashtag related to that topic. She uses Twitter advanced search and Hootsuite [111] to find such accounts, and Storify [167] and Banjo [16] to generate stories and identify people to interview. However, SME1 stated that the tools she is currently using miss relevant information: "*we tend to fall back to those things that are easiest often because we are rushed, we follow an account, or we follow a hashtag, but we do miss a whole lot. Because there are relevant tweets that do not have those keywords and that has always been a problem*".

SME2 has extensive experience in performing content analysis or discourse analysis on traditional media such as news articles. However, she has not used any content analysis tools, but manually studied the news articles, as in her opinion, content and discourse analysis tools are not rich enough to always capture the meaning of the

article. She added that ensuring that important information is not missed is a very difficult task and an issue for social science research. SME2 plans to perform content and discourse analysis on Twitter for her research, and she knows that she cannot rely only on some keywords for finding relevant information.

SME3 has made use of Twitter advanced search and stated that its results are not very accurate: "*It gives you tweets that does not have anything with what you are looking for*". She added that in her opinion this search engine also misses relevant tweets. She supports her opinion with an example about how hashtags can result in receiving irrelevant information. During the last winter Olympics, the hashtag "#WeAreWinter" was used in Canada in tweets supporting Canadian teams or reporting news about these games. However, some people used this hashtag in posts not related with the Olympics. For instance, somebody might just say "I just went to the supermarket #WeAreWinter".

During Pair Analytics with ATR-Vis: Main Interactions

SME1 found that "*the tool is pretty straightforward*" to use and that showing the discriminative features are useful especially because the user can control their assignment to different debates. She also mentioned that the Similarity Hashtag-Debate panel in the More view is useful for labeling all tweets containing a specific hashtag with one single assignment, which in turn may reduce the number of labeling requests as well. In addition, in her opinion, the Force Layout View is very helpful in determining the clusters of tweets. Therefore, the user can perform a deeper analysis on these clusters and see whether a story exists or not. For instance, SME1 commented that there is a dense cluster for the "Fair Elections Act" and she would be interested in analyzing this cluster to determine whether "*one political party or one political group is really responsible for a lot of this conversation*". She also mentioned that the Force Layout can be used for determining debates with a broader range of topics from the composition of the clusters. She exemplified her remark with the observation that debate "Aboriginal Affairs" seems to have a broader range of topics, which are in common with other debates, as compared to "Fair Elections Act".

SME2 liked all the features and mentioned that the Assignment View is "*very*

useful and user friendly". She found particularly useful that she can assign discriminative features to different debates and see the effect of this assignment on how tweets are retrieved. She added that: "*people make a lot of bizarre references to things that have nothing to do with something else*" and therefore it is important that ATR-Vis gives the ability to examine tweets retrieved by the automatic method and change the debates/topics of tweets. She mentioned that such a feature is very useful when the computer or the user makes a mistake: "*It alleviates human error ... in social sciences*". Also, by looking at different branches of a conversation and the colors of its nodes in the Reply Tree, she noted how fast people can change their minds about a topic.

In SME3's opinion all features of ATR-Vis are useful: "*I think it is all really useful and what part becomes the most useful depends on the individual topic*". For instance for some topics, the Reply Tree will be extremely useful, but with other topics, it might not be as useful as visualizations in the context view. Then, she continued "*I do not think that there is anything on here that is a waste of space and it is all useful*". She also found the ability to make corrections even to her own errors, with simple interactions such as drag and drop, very helpful.

After Pair Analytics Session: Questionnaire

We posed three main questions to SMEs after finishing using the tool: "What advantages and possible other uses you find for ATR-Vis?", "Would you consider using this system for your own work/research?" and "What limitations did you find and/or what suggestions can you give us to improve the tool?"

What advantages and possible other uses you find for ATR-Vis?

SME1 mentioned that ATR-Vis can be useful for other tasks that go beyond the goal of just searching for tweets. The first task is to find active Twitter accounts in her topics of interest: "*I can see this being useful at various points along the process for a journalist, one is looking for people...When you are assigned to a story and you are doing background information, so one way would be to find people. Because if you find people who are actively engaged on Twitter, you can track them down, you can call them up, you can do interviews*". The second task is to learn about emerging topics and events. The SME currently utilizes Google News alerts to receive information

about her topics of interest, but she mentioned that it searches for news stories only, not tweets. The last task is to look for patterns and trends and identifying related debates/topics from clusters of tweets that look interesting for story ideas; e.g. "Marine Mammal Regulations" may be related to "Employment" considering the Inuit communities. She also exemplified this point saying: "*just looking at the connection between Nigerian girls and missing and murdered indigenous women, ... people are kind of putting the two of them together; that could be a news story*". SME1 also highlighted the serendipity allowed by ATR-Vis: "*Sometimes we don't know what it is we are looking for and sometimes it's like you have a hypothesis, so I think I know what my story is about, but I can ignore the evidence or I can ignore what is in front of me and I have to rethink my story and my focus and then reassess. So, a tool like this is great at every step of the game of the story*".

SME2 pointed out that showing the similarities between hashtags and debates in the Similarity Hashtag-Debate panel is useful for the content/discourse analysis as only considering the appearance of hashtags in tweets cannot always capture their meaning: "*in criminology people use a lot of hashtags specially with race, issues, fear and crime in general*". SME2 added that ATR-Vis can be used in identifying the connections that people make: "*in my line of research that is what makes it important and it is something that I could never do on my own and that's what makes a program like this so important is to actually look at the verbal connections that people make on their own*".

SME3 commented on the possibilities for ATR-Vis to gather accurate public discourse: "*I definitely stand by thinking that it's going to be the best way to get really good public discourse on issues in any social science*". She added that performing traditional research in social sciences, through face-to-face interviews or questionnaires, has many difficulties such as finding people who are willing to be interviewed in today's fast-paced world and also avoiding social desirability bias. In her opinion, ATR-Vis can help social science researchers gather more accurate opinions faster and easier than traditional methods in social sciences, which is very important in their line of research. She concluded saying: "*ATR-Vis is far more accurate than using Twitter Search*".

Will you consider using this system for your own work/research?

SME1 stated: "*I would definitely try this again*" and "*I would even have this as one of the tools in our students' toolbox to use when they are working on their stories*". SME2 commented "*This has actually exceeded all of my expectations because it just makes the possibility of my research big*". She added that the research possibilities are endless and the fact that there is a system that can make it happen is interesting. The SME mentioned that, although she is relatively new to the study of social media, she finds ATR-Vis very useful: "*This is something that I would use for every single piece of research, something that students can do master theses on*". SME3's response was: "*Yes, it is definitely very user friendly and well designed*".

What limitations did you find and/or what suggestions can you give us to improve the tool?

Both SME1 and SME2 commented that they miss the capability of adding new debates on the fly to the list of debates. For instance, SME1 mentioned that she found interesting tweets and discussions about the debate on abortion among the tweets and wanted to add this topic/debate to the list in order to retrieve its relevant tweets.

SME1 added that being able to put different tags/notes on tweets, hashtags and features would be very useful, especially when multiple users are working together or the user is interacting with the system in multiple sessions. For instance, the user may put some tags showing how certain she is about the label of tweets and hashtags or even about the authority of an account. In this case, she may want to further investigate cases with low certainty or ask her colleagues' opinions about them. Finally, integrating ATR-Vis with Facebook posts and Instagram is another addition to the system commented by SME1. She also mentioned that being able to track back stories and see when they started and what was the trigger, i.e. having a timeline, would be useful.

3.4.6 Discussion

We discuss various aspects of the experiments described above in this subsection.

Numerical Experiments

The experimental results obtained on two distinct datasets showed the advantages of applying our selection strategies. In general, the poorer the retrieval for a given debate

is, the more it benefits from the active retrieval strategies for improving its retrieval results. The selection strategies based on the ambiguous retrieval and hashtags are the most effective ones as shown in Tables 3.3 and 3.5. However, the strategy of simulating the user introduces certain limitations. For the Canadian dataset we leveraged 12 hashtags, which generate 36 requests as we assume that 3 tweets are needed to inspect the hashtag. More realistically, in view of her domain knowledge a user may not need to inspect a hashtag to understand how it is used. Likewise, the reply strategy is more suitable for visual inspection rather than for a massive retrieval after some random posts in the reply chain are inspected.

Sensitivity due to Data Sampling

We also analyze how sensitive the experimental results are to the specific sample considered by investigating how they are affected when more labeled tweets are added. Therefore, we first considered only subset B, which contains 60% of the retrieved tweets for different debates, as our test set and then we evaluated the effects of adding to it the 1,000 randomly selected tweets of subset A. Precision remains stable for most debates as it is observed in Fig. 3.12. For debates with relatively few tweets such as "Local Food" and "Palliative and End of Life", which are more sensitive to small changes in their retrieved tweets, there is a reduction in their precision. This also explains the low precision of the retrieval methods (Table 3.4), which results from retrieving a few wrong tweets for these debates. For recall, the effect of adding subset A into subset B is even less significant. The comparison of the overall accuracy ($B = 0.98, B \cup A = 0.99$), macro-precision ($B = 0.82, B \cup A = 0.83$) and macro-recall ($B = 0.84, B \cup A = 0.85$) also indicates no major effect of extending the test set with randomly sampled tweets.

Model Selection

The results reported in Tables 3.3 and 3.4 show the accuracy of our methods for the parameter settings introduced in Section 3.4.2. Although the performance of the method may be sensitive to different parameter configurations, we hypothesize that the proposed active retrieval strategies can compensate for negative effects incurred due to parameters not set optimally. The accuracy of our unsupervised and ATR

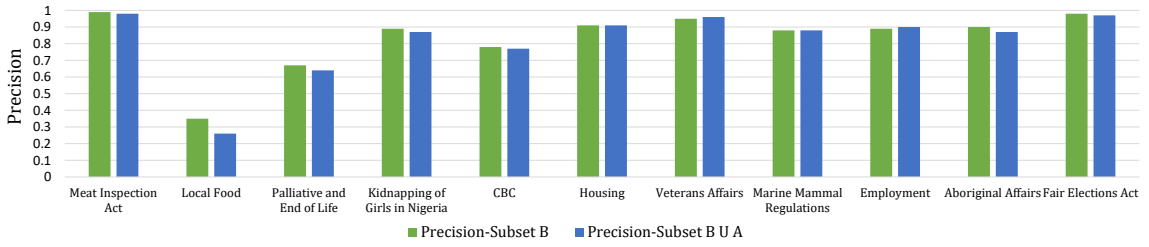


Figure 3.12: Precision for each debate before and after adding tweets of subset A, i.e. 1000 randomly selected tweets, to subset B, 60% of the retrieved tweets for different debates.

methods in terms of the number of keyterms used in the initial step, κ, is presented in Fig. 3.13. The solid blue line shows the accuracy for the unsupervised retrieval model, where one observes that performance is affected by the choice of this parameter. The black-dotted line represents the result after the ATR strategies, and we can see that these strategies improve the results of the unsupervised retrieval fairly independent of the parameter choice. The solid orange line shows the results of ATR after doubling the number of labeling requests. The result supports our hypothesis that additional requests to the user can compensate a non-optimal parameter configuration, as in all cases a similar upper bound in accuracy is obtained regardless of the number of keyterms used.

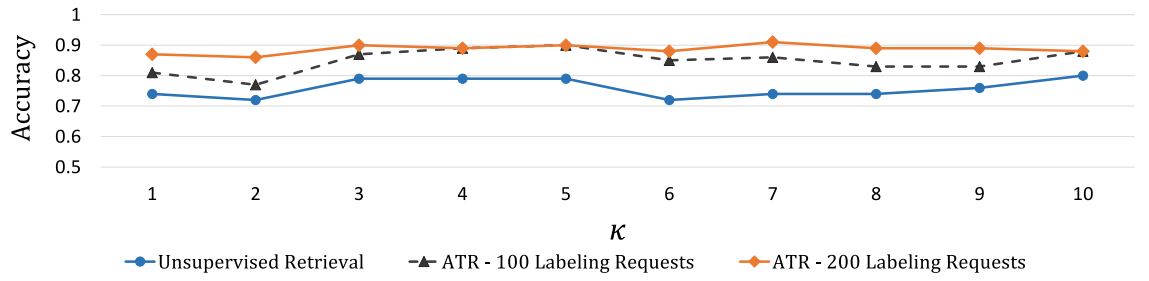


Figure 3.13: Accuracy before and after applying ATR strategies with different values for the number of keyterms. A value of $\kappa = 5$ is used in our experiments.

While developing the active retrieval strategies we tested several other approaches. This involved different keyterm extractions for debates and tweets, different text similarities, different vector-space text representations (e.g. character and word-bigrams with binary, integer and real-valued weights) and different approaches to identify stop hashtags (e.g. degree centrality on the co-occurrence graph). We selected the

design options that worked best for our tasks, while we favored the simpler model or approach when no clear difference was observed.

Use Cases

The debates in the Brazilian dataset are more semantically related to each other than the Canadian one, and hence their retrieval performance is lower. Furthermore, the week of May 2015 was shaken by the news of investigations on corruption and arrests of high-level managers in the Federal International Football Association (FIFA). Given that football is a very popular and sensitive topic to many Brazilians, this resulted in a large number of noisy posts being merged with the discussions of the ongoing debates at the Brazilian Senate. Yet, this allowed us to evaluate ATR-Vis in the context of uncontrolled external events.

Regarding the use cases of parliamentary datasets, some interesting differences were observed between the two datasets. The typical reply pattern in the Brazilian dataset is the public replying to senators without interacting among each other, while in the Canadian dataset people tend to engage more in conversations. In the Brazilian dataset we also noted that tweets involving the senator Romário de Souza Faria—who used to be a highly popular football player—seem to generate a substantially higher number of reactions among Brazilians compared to any other senator. In the case of the Canadian Parliament, we note that the leaders of the major political parties are the ones that trigger most of the tweets followed by those senators involved in the proposal of the bills of interest.

Retrieval Error Analysis

The proposed automatic retrieval method is not flawless, which indeed prompted our motivation for incorporating the user into the retrieval loop. This is particularly the case at the beginning where some initial features may not be good indicators for discriminating the debate of interest, so user intervention is necessary. The correct identification and filtering of hashtags by the system plays a major role in the retrieval of tweets. False negatives in the identification typically leads to a large number of tweets being retrieved to the wrong debates, or failing to retrieve those. False positives are less harmful but imply in unnecessary labeling effort from the user.

Inadvertent user mistakes when providing feedback are also possible. In some scenarios, some of the visual components of the tool can be useful to detect such inconsistency. For instance, if a user mistakenly assigns hashtag "#c23", which is the bill number for "Fair Elections Act", to the debate "Marine Mammal Regulations", numerous connections between these two debates in the Ring Visualization provide an indication that there has been an error in the retrieval/supervision, especially if those connections are unexpected for the user. Subsequent inspection of the tweets and their discriminative keywords can fix the incorrect supervision.

However, for some cases neither the system nor the user could identify possible inconsistencies, and hence incorrect retrieval can pass unnoticed. Most of these errors result from difficulties in understanding the semantics of tweets, as the method mostly relies on vector similarities. Sarcastic tweets are one common example. Another example is spam tweets that use misleading URLs or hashtags just for the sake of being visible among trending topics.

3.5 Conclusions

We presented ATR-Vis, a user-driven visual framework for active retrieval targeted specifically for Twitter data. This framework addresses an existing challenge in the analysis of social media data, which is to assure that the information relevant to an analytical task is retrieved by attempting to maximize both recall and precision. The proposed framework has been applied and evaluated in a task scenario of retrieving Twitter posts related to a set of target debates occurring in a parliament house over a certain period.

The experimental results demonstrate that the framework can successfully integrate a user into the retrieval task so as to improve both retrieval precision and recall. User involvement is kept to a minimum by carefully selecting and submitting Twitter entities, i.e. posts or hashtags, for user supervision based on their estimated potential to improve the retrieval outcome. Our proposed strategies for selecting these entities were favorably compared against other approaches including a state-of-the-art system for interactive retrieval.

The interactive interface enables a user to explore, inspect and adjust the retrieval

process, so that user interactions actually modify how the system works. It gives non-technical users—who might be political analysts or journalists—the tools for obtaining a reliable Twitter collection responding to their information needs before carrying out any data analysis. Furthermore, this user-driven approach yields a higher versatility to adapt the framework to different domains without any additional model refinement, which would typically require a data mining expert. We showcased possible flows of analysis using ATR-Vis in the context of three datasets. No language tuning was required in handling content in English or in Portuguese, as all methods implemented in the framework are language independent.

In order to have a stronger understanding of ATR-Vis, its different interactive visual features, and its efficiency in assisting users in finding relevant information, we performed an evaluation with three domain experts. All three experts provided positive feedback for ATR-Vis, and acknowledged the need for this type of tool for the accurate retrieval of tweets. They also shared interesting insights on potential improvements and further developments.

3.5.1 Limitations and Future Work

There are several possible avenues for extending this work. We list them below.

1. Evaluating ATR-Vis framework in the context of stream data—as opposed to the static dataset evaluations considered in this thesis. This will involve finding strategies to see how the system faces the cold start problem at the beginning and how it should "forget" data when memory is exceeded or older keywords do not apply anymore.

2. Assigning impact scores to each labeling request based on its probability of improving the overall accuracy of the retrieval. It can help users in assessing the labeling effort and deciding which tweets to label.

3. Incorporating other context-based and social features in the retrieval process. ATR-Vis takes into account only terms from the text of the tweet. Other context-based features, such as length of the tweet and number of affective words in the tweet, as well as social features, such as number of followers of

the author of the tweet and number of her published tweets, could be considered. Appendix F illustrates the distribution of relevant tweets over time. We could not observe a direct relation between the time tweets were published and their relevance to particular debates. However, it requires further analysis by incorporating such features in the retrieval process.

4. Expanding tweets with the text of documents referred to by the URLs in the tweets. Investigating the impact of enriching tweets with the content of these documents on the accuracy of the results is another avenue of future work. However, it is important to identify spam URLs first as they can deteriorate the performance of the retrieval.

5. Considering the case where the user can make mistakes in the labeling process inadvertently. Although we expect that incorrect labels will decrease the retrieval accuracy, it would be useful to study the robustness of the ATR-Vis approach.

6. Analyzing concept drift in conversations. While we assumed that tweets in a reply chain should be associated with the same debate as their source tweet, if the reply chain becomes very long, the topic of discussion may drift. It would be interesting to examine the potential effect of the length of the reply chain on the consistency of their topics and subsequently on the effectiveness of the selection strategy.

7. Assuming tweets can be associated with more than one debate. Since tweets are short, we presumed each tweet is related to at most one debate. Further evaluation on the accuracy of this assumption for closely related topics is required. In case of the assumption that each tweet can be retrieved for more than one topic, it would be interesting to investigate multi-label active learning techniques, which take into account the probability of instances being relevant to multiple topics at the same time in their selection criterion. In addition, it has been shown that for multi-label active learning scenarios, where an instance can be related to more than one category, types of requests are more important than the selection strategies [119]. For instance, evaluating the algorithms that

ask the user to order different categories based on their relevance to the instance in the labeling request would be an interesting research direction.

8. Evaluating the level of transparency and user control. ATR-Vis is transparent about its retrieval process and helps users better understand how tweets are retrieved for different debates. In addition, users have control over this process by different interactions, such as assigning discriminative features to different debates and simply selecting any tweet as a labeling requests. Further studies will shed some light on whether providing more or less transparency and control to the user improves or deteriorates the performance of the retrieval process.

Chapter 4

Microblog Filtering based on User Interest Profiles

4.1 Introduction

Twitter is the most common microblogging platform with millions of users, who broadcast short messages, which are known as tweets, and read other users' messages to obtain recent and novel information about current affairs. For instance, journalists use Twitter to find worthy news and track sources [182]. Many users collect valuable information about their areas of interest, such as ongoing events, professionals in those areas and relevant organizations. Meanwhile, the vast number and variety of discussed topics and shared information makes it difficult for users to filter relevant and non-redundant tweets about their topics of interest. Information filtering and recommendation systems, which have recently attracted much attention, assist users in this process by presenting to the user only the information that is relevant to them [20, 89, 232, 104].

There are different challenges when working with Twitter data including the noisy characteristics and dynamic nature of its content. Tweets are known to be of a noisy nature and lacking context due to their brevity [238]. In addition, tweets are typically interesting only for a short time after they are published. Furthermore, since new topics and discussions emerge every day and users change their interests frequently, it is difficult to obtain labeled data to learn from. These characteristics make tweet recommendation a challenging task.

Traditional information retrieval systems such as those based on language models have been used in information filtering systems [302, 207]. However, considering the challenges of processing Twitter data, the same level of performance for these methods cannot be expected. Based on our goals and objectives and the proposed framework in Section 1.3, we aim at involving a human or information source who, by providing minimal supervision, can overcome most of the above challenges to significantly improve the performance of tweet filtering and recommendation systems.

4.1.1 Research Problem

In this chapter, we address the task of finding novel and relevant tweets based on users' interest profiles, which can be divided into two parts: deriving a query from each user interest profile, and ranking tweets based on their relevance to the query. Our research problem assumes that textual descriptions of users' interests are available, which is different from typical recommender systems, where the systems estimate users' interests from their history (e.g. items they liked in the past) or from the interests of other similar users. On the contrary, in our problem setting users explicitly express their interests. We consider the task introduced in the TREC 2015 Microblog Track (Scenario B, also referred to as email digest) [158], where filtering systems retrieve up to 100 tweets for each user interest profile per day, as the settings for our evaluation. Retrieved tweets should be ranked based on their relevance to the given profile and should contain novel information about the user interest. We refer to this scenario as TRECMicroB for the rest of this document. Our research problem can be defined more formally as follows:

Definition 6 *Given a time-ordered stream of Twitter messages and a profile p, where the user explicitly expressed her interests, at the end of each day d, the task is to retrieve a ranked list* $R_d = \langle t_1, t_2, ..., t_{|R_d|} \rangle, |R_d| \leq l$*, where* l *is the maximum number of recommended tweets per day, such that for any pair* $(t_j, t_{j'}) \in R_d$ *with* $j < j'$*,* t_j *is more relevant to profile p than* $t_{j'}$*, and* $t_{j'}$ *should contain novel information not considered in any previous tweet of the list.*

We intend to perform the task of tweet filtering automatically and then improve the performance of the system by incorporating an information source. The information source can provide the relevance of a tweet, also referred to as label, at a defined cost. Considering the large number of tweets published every day and the cost of labeling, selection strategies become crucial. We consider three levels of relevance as the valid labels: not relevant, somewhat relevant, or highly relevant. We use "relevant" to refer to both somewhat relevant and highly relevant labels.

Our approach is to use information retrieval algorithms, which retrieve and rank matching documents based on their relevance to the query. One of the main challenges of information retrieval methods in the context of Twitter data is the vocabulary

mismatch between the content to be retrieved and the formulation of the query. Expanding the query is one of the most common techniques to address such challenges. Different automatic query expansion methods, such as pseudo-relevance feedback, have been proposed for tackling this problem and they have shown to be successful in improving the results for web documents [36]. For instance, semantic relatedness models can be used for finding the most related words to the query terms, which are in turn added to the query [94]. However, query expansion methods should be used carefully as they can deteriorate the precision of the results [36].

We first present a framework for Twitter Information Filtering (TIF) based on user interest profiles. Then, we use semantic relatedness methods to automatically expand the query and evaluate these techniques in improving the performance of the tweet filtering systems. Finally, we propose active learning strategies for modifying the query based on the provided labels by an information source and analyze the effectiveness of the proposed selection strategies on the performance of our filtering system.

The proposal of our TIF framework includes the following research contributions:

- The evaluation of state-of-the-art semantic relatedness models for query expansion and the analysis of different ways by which query terms can be expanded.
- The proposal of a novel active learning strategy for tweet filtering systems based on query expansion and semantic relatedness methods.
- The demonstration of the effectiveness of our proposed techniques by filtering 16,302,498 English tweets for 51 profiles and the comparison with a state-of-the-art learning-to-rank method under several sampling strategies.

4.1.2 Overview

This chapter addresses the task of tweet filtering based on user interest profiles. After reviewing related work on tweet filtering and recommender systems in Section 4.2, we propose an unsupervised method, which returns a list of tweets ordered based on their relevance to a specific user profile in Section 4.3.1. We then discuss automatic query expansion in order to improve the performance of our filtering system in Section 4.3.2. Following our proposed framework in Chapter 1 for involving the user in the process, we propose active learning techniques for improving the quality of the results by

expanding the query in Section 4.3.3. Analytical results in Section 4.4 demonstrate the effectiveness of our proposed strategies in increasing the gain from user supervision, while also outperforming one of the best supervised learning-to-rank methods. The conclusions of this chapter is presented in Section 4.5. Fields of a tweet that are referred to in this chapter are also summarized in Appendix E.

4.2 Related Work

Twitter users, even those following a few users, can be overwhelmed by the high volume of tweets. Several approaches have been proposed for overcoming this information overload problem such as categorizing users, classifying tweets and employing learning-to-rank methods. In addition, techniques such as query expansion and semantic enrichment of tweets were proposed for alleviating the challenges of processing tweets.

A group of methods employ techniques to categorize Twitter accounts that a user is following for different purposes [227] in order to facilitate the association of relevant tweets to each of her interest topics. In addition, finding authoritative and influential authors for a given topic has been studied in other works [210, 287]. After identifying these leading authors, their tweets are suggested to other users that are interested in the same topic. However, sometimes a user is following another user because they both share an interest, but their whole set of interests may not be necessarily the same [85]. For instance, if user A is following user B because of her technology-related tweets, A may not be interested in B's tweets about art and sport. Therefore, recommending to user A all of user B's tweets may not be an effective approach. Also, a graph-based approach for recommending lists, groups of Twitter accounts, to users was proposed in a previous study [228], where users can subscribe to lists that they find interesting in order to receive relevant information and be united with users with similar interests.

Another group of methods identify relevant tweets to user interest profiles by classifying tweets. A system for filtering breaking news from noisy tweets by classifying them into two categories, junk or news, is TwitterStand [248]. One of the first research studies on improving the performance of tweet filtering systems was performed by Sriram et al. [257]. They showed the limitations of traditional methods

when applied to short text messages and proposed a method that uses features extracted from the authors' profiles to classify tweets into predefined generic categories, such as news, events, and opinions. On the contrary, the user profiles considered in this thesis contain custom information needs, which require a more precise filtering process. Moreover, besides filtering relevant tweets from irrelevant tweets, we aim at recommending tweets in a ranked list based on their level of relevance.

Grouping tweets in a user's feed based on their topics, instead of presenting them in a chronologically-ordered list was proposed in Eddi [19]. The evaluation performed by active users showed that Eddi is more efficient and enjoyable than the standard chronological interfaces. Recommending friends' activities from different social networking sites was proposed in SocConnect [304]. This interactive system enables users to aggregate and organize their social data from different social media platforms, such as Twitter, Facebook [68] and LinkedIn [159], into a single location. Users are able to map their friends' accounts across different sites and also create groups of their friends. In addition, after users rate their friends and their activities as favorite, neutral or disliked, a machine learning classifier is trained on the provided ratings. Then, new activities of the user's friends that are classified as interesting by the trained model are recommended to her.

Twitter also offers filtering services, where users add some keyword-based filters and the filtering service returns matched documents from the stream [241]. These filtering services have access constraints. For instance, Twitter has limited the number of keyword-based filters to 400 and the maximum number of returned documents is around 1% of total tweets' rate. In addition, this way of retrieving relevant tweets is biased towards provided keywords. In a recent work, keyword clustering techniques were proposed for improving the coverage of the retrieved tweets by Twitter filters [247].

Learning-to-rank methods have been employed in tweet filtering systems [276]. For instance, SVMRank was applied to three types of features including content relevance, Twitter-specific, and account authority features [64]. The evaluation setting of that work is quite different from ours as its dataset is considerably smaller and about half of the tweets are relevant to the queries, which is much higher than the percentage of relevant tweets in our dataset (less than 0.05%). It was shown in a user study

that there is a correlation between the tweets that users are likely to retweet and the ones that they are interested in reading [273]. Four feature categories (tweet-based, user-based, content-based and author-based), and the Coordinates Ascent learning-to-rank algorithm were used to rank tweets based on their probability to be retweeted by users, or in other words based on their degree of interest. In addition, Gradient Boosted ranking was used to reorder tweets in a user's feed based on her interests, where the recency of tweets, influence of authors, content of tweets, and social features were included as features in the model [252].

Query expansion and enriching tweets with external knowledge sources have been used for addressing the aforementioned challenges of Twitter data. A dynamic query expansion approach for retrieving microblog posts, which takes into account the time factor and includes Twitter-specific features such as usernames, hashtags, and links was proposed by Massoudi et al. [184]. In another work, queries were expanded with the incremental Rocchio algorithm to overcome the sparsity of user interest profiles [8]. There is also a recent research study on the effects of relevance feedback and query expansion on the performance of information filtering systems [17]. External knowledge sources, such as Wikipedia [288] have also been used for tackling the sparsity problem of short texts [87]. For instance, Wikipedia and WordNet [223] have been integrated with the clustering of short texts [114]. In addition to expanding tweets with lexical features obtained from knowledge sources, the graph structure of knowledge sources, e.g. DBpedia [53] and Freebase [95], has also been used for the topical classification of tweets [33].

It is worth mentioning that there are two common types of recommender systems: 1) Collaborative Filtering, where the recommendation is based on the history of the user and her like-minded users [42], and 2) Content-based, where the recommendation is based on the description and properties of items and user interest profiles [219]. Hybrid recommender methods use a combination of the collaborative filtering and content-based approaches [41, 40]. When users do not explicitly specify their interests, users' interests are implicitly inferred from their previous published tweets [2, 137]. For instance, categorizing users' tweets containing URLs by classifying the contents of their referred webpages to 18 general topics was proposed as a user profiling approach [85]. In this case, the topics are used to prioritize relevant

information in users' Twitter feed. For passive users, with only few or no posted messages, tweets published by users' followees are useful to infer interests of these passive users [216]. We focus on content-based approaches for cases when users explicitly specify their interests. A taxonomy of recommendation tasks in Twitter and the existing methods and techniques in each category was also presented [150].

Interactive recommendation and information filtering systems that benefit from user involvement have been in used for the past decade [168, 265, 284]. It has been shown that visualization techniques are beneficial in interactive recommendation systems. For instance, TasteWeights [24] is a visual interactive hybrid recommender for music, while PeerChooser [204] and Smallworlds [97] present interactive visualization systems for movie recommendation. All these systems present interfaces that explain the recommendation process in a transparent manner and is also used for collecting the relevance feedback from users about the recommended items. User studies for evaluating these systems demonstrated that a visual interactive interface helps improve the performance of the recommendation systems and enhance user experience with the systems.

There are some recent studies on interactive systems for microblog platforms. A recent study proposed a recommendation system that instead of filtering out irrelevant tweets, it draws users' attention to more recent, relevant and interesting tweets [277]. The proposed visual interface [278] enables users to control over their stream consumption, while no tweet is removed from the stream. The pilot study demonstrated that users trust systems that do not filter any activity in their network and give them more control over what is presented to them.

Twitcident is a framework for filtering and analyzing tweets related to real-world events [3]. The authors proposed a semantic enrichment of the tweets by applying named entity recognition, classification of the textual content of tweets, and extraction of further meta-data from the tweets and the webpages pointed by URLs contained in tweets. They showed that these semantic enrichments improve the performance of their filtering and search system. Twitcident provides an analytical tool that enables users to explore the data and interact with the system based on their information needs, but the system does not take advantage of any active learning techniques. TweetTracker [149] is a monitoring and tracking tool designed for humanitarian aid

and disaster relief respondents to help them gain real-time awareness of situations in disasters. This tool filters tweets from temporal, geo-spatial, and topical perspectives and provides a web-based visualization module for user interaction. These types of systems are often designed for particular users, which is different from our defined task in this chapter.

Considering the problem settings of tweet filtering task and the dataset used in experiments, two recent research studies by Tan et al. [261] and Zhu et al. [309] are the closest to our work in this chapter. Both studies used the dataset from the TREC 2015 Microblog Track [158], but their reported results are related to a different scenario of the Microblog Track, i.e. Scenario A (mobile notification scenario), where systems should notify the user about interesting tweets shortly after being published rather than at the end of the day, and thus are penalized for latency in their notification. Tan et al. used relevance feedback for determining a dynamic threshold, which verifies the relevance of tweets to different user interest profiles. Considering daily relevance feedback, they determined the optimal threshold for each profile, which showed improvement on the performance of their filtering system. This approach is similar to our work considering the user involvement, but no particular strategies are used for selecting the most appropriate labeling requests. In addition, they apply query expansion, which is performed automatically using pseudo-relevance feedback, while our query expansion is based on the labels of the selected labeling requests.

The tweet filtering approach proposed by Zhu et al. is based on semantic expansion of tweets and profiles in addition to a tweet quality model that identifies tweets with high information value using social features such as the number of author's followers. The incorporation of semantic features is performed by employing a word embedding model. They also developed a boolean logic keyword filter for verifying that a tweet is relevant to the target profile, and utilized external search engines for query expansion. Our work differs from their study in several aspects with regard to the techniques used for tweet retrieval, relevance verification, and query expansion; more importantly, their proposed method dose not consider user supervision and thus no active strategies are employed.

We use an information retrieval method for finding relevant tweets and propose a query expansion method to increase the number of retrieved relevant tweets. Our

query expansion method is based on the semantic relatedness methods and a set of strategies that actively select tweets as labeling requests. Using semantic similarity methods for finding similar phrases in tweets in order to remove redundant tweets has been proposed [295, 301]. In a recent work, word co-occurrences were used for computing the semantic relatedness between two terms on Twitter and it has been discussed that corpus-based semantic relatedness methods might be more suitable for Twitter data than techniques that are based on knowledge sources such as WordNet [72]. Studying semantic relatedness methods is a different research problem, which is out of the research scope of this thesis. We use existing semantic relatedness methods in expanding the query with related terms in order to find more relevant tweets. To the best of our knowledge, active query expansion with semantic relatedness methods have not been studied for the tweet filtering systems. We also compare our proposed method with a supervised method from the learning-to-rank category, SVMRank.

4.3 Proposed Method

We first describe our general framework, TIF, and its components. Then, we discuss automatic query expansion and our proposed strategies for active tweet filtering, ACTIF.

4.3.1 Unsupervised Tweet Filtering

Our framework for Twitter Information Filtering (TIF) consists of several components, including query formation, tweet retrieval, tweet relevance verification, novelty verification and final relevance ranking, which are shown in Fig. 4.1 and described in the following sections. The notation used throughout this chapter is also summarized in Table 4.1.

Query Formation

User interest profiles in TRECMicroB contain a title, a description and a narrative. However, a more realistic scenario is to assume that only a title exists, or that users specify the gist of their interest in a few keyterms or by providing a short query.

Table 4.1: Notation used in this chapter

Notation	Description
T	Tweet stream
t_j	A tweet in stream T
$t_{j,o}$	A term in tweet t_j
p	A user interest profile
$\widetilde{Q}$	Set of query terms initially taken from the title of profile p
q_i	A query term in $\widetilde{Q}$
E	Set of named entities in $\widetilde{Q}$
S	Set of semantic groups for profile p
s_i	A semantic group, i.e. a set of semantically related terms to q_i
$s_{i,r}$	A term in s_i, i.e. q_i or any of its semantically related terms
w_i	Weight of s_i, which indicates the importance of s_i for profile p
Q	A query constructed by inclusion/exclusion of semantic groups
$label(t_j)$	A function that returns the actual relevance of t_j to p
R_d	Ranked list of relevant tweets to profile p at the end of day d
R	Set of all recommended tweets by the filtering algorithm for p
$rel\_score(t_j)$	A function that returns the relevance score of t_j to profile p
$sim\_term(t_{j,o}, s_i)$	Semantic relatedness value between term $t_{j,o}$ and s_i
$sim\_tweet(t_j, S)$	Semantic relatedness value between tweet t_j and set S
$\mathbf{t_{j,o}}$	Vector representation of term $t_{j,o}$
$cos(\mathbf{t_{j,o}}, \mathbf{s_{i,r}})$	The cosine value between vectors of $\mathbf{t_{j,o}}$ and $\mathbf{s_{i,r}}$
$rel\_term(t_j, s_i)$	A function that returns a term in t_j as the semantic relatedness term to s_i, or returns NULL if such a term does not exist
C_d	Corpus of tweets published in day d

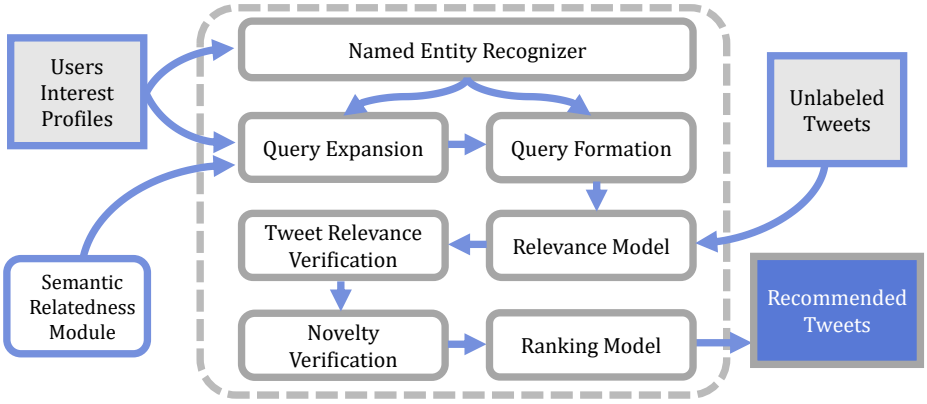


Figure 4.1: The proposed framework for filtering tweets based on user interest profiles

Therefore, for each profile, we construct a set of query terms from the terms of its title by removing any stop words and applying stemming to the query terms.

Let $\widetilde{Q} = \{q_1, q_2, ..., q_n\}$ be a set of query terms that are initially taken from the title of profile p. Then, for each query term q_i, we define a semantic group s_i as follows:

Definition 7 *A semantic group s_i is a set of semantically related terms to the query term q_i and including the query term itself, $s_i = \{s_{i,1}, s_{i,2}, ..., s_{i,m}\}, 1 \leq i \leq n$. Each semantic group initially contains the query term only, and as we continue with the proposed query expansion strategies, other semantically related terms are incrementally added to them.*

The idea of each semantic group is to include synonyms of the query terms that can be used to increase the recall of the recommended tweets. We also assign a weight to each semantic group which indicates its importance in conveying the meaning of the user interest. Therefore, we define $S = \{(s_1, w_1), (s_2, w_2), ..., (s_n, w_n)\}$ as the set of semantic groups, where w_i is the associated weight with the semantic group s_i. These weights are used in verifying the relevance of tweets to the profile (see Subsection "Tweet Relevance Verification") and also as a boosting or damping factor in ranking matched documents with the query (see Subsection "Tweet Retrieval"). We consider two different strategies for assigning weights to the semantic groups. Our initial model assumes that all semantic groups are of the same importance in representing the user interest profile, so it assigns equal weights to all groups in a way that their sum is equal to one, i.e. $w_i = \frac{1}{|\widetilde{Q}|}$.

Our second strategy for assigning weights to semantic groups is based on the hypothesis that named entities convey a high degree of specificity. Therefore, we assume that semantic groups containing named entities are more important than semantic groups not containing named entities in representing the user interest. More specifically, our assumption is that a relevant tweet must contain all the named entities, while occurrence of one or more semantic group not containing a named entity is enough. Consequently, a value equal to the weight of semantic groups not containing named entities is distributed among semantic groups containing named entities. We first identify named entities in the query by applying the Stanford Named Entity Recognizer (NER) [75]. Then, we use the following equation

$$w_i = \begin{cases} \frac{1}{|\widetilde{Q}|}, q_i \notin E \\ \\ \frac{|\widetilde{Q}|-1}{|E||\widetilde{Q}|}, q_i \in E \end{cases} \tag{4.1}$$

for the assignment of weights to each semantic group, where E is the set of all named entities in the set of query terms, $E \subseteq \widetilde{Q}$. If $\widetilde{Q}$ only contains named entities, then the weight of each named entity is equal to $\frac{1}{|E|}$. We refer to this second weighting strategy of our proposed method as NEI (Named Entity Importance). The named entities that we consider come from the Stanford 4-class model [75], which are locations, persons, organizations, and miscellaneous.

In different steps of our proposed framework, TIF, we create queries from the set of semantic groups. For instance, $Q = \bigcup_{s_i \in S} s_i$ is a query that contains all the terms in all semantic groups. Since we initialize each semantic group with its corresponding query term, i.e. $s_i = \{q_i\}$, our initial query is the union of all query term $q_i \in \widetilde{Q}$.

Tweet Retrieval

Information retrieval models that retrieve a ranked list of documents for a given query are directly relevant to filtering systems [18]. We consider a probabilistic language model [51] as the core of our tweet retrieval model. At the end of day d, all tweets that are published in that day are considered to estimate a language model for each tweet. Then, tweets are ranked by the probability that the query would be generated by their model. Most language models used in information retrieval and filtering tasks

are unigram models, since they are often sufficient to judge the topic of a text and also more efficient to estimate and apply than higher-order models [180]. The assumption of the unigram language model is that each query word is generated independently, i.e. $p(Q|t_j) = \prod_i p(q_i|t_j) = \prod_i (\frac{tf_{q_i,t_j}}{|t_j|})$, where $p(Q|t_j)$ is the probability of the query being generated given the tweet t_j, and tf_{q_i,t_j} is the occurrence frequency of query term q_i in t_j.

Based on the maximum likelihood estimator, the probability of any unseen word in a document is zero. Therefore, different smoothing techniques are applied to assign a non-zero probability to the unseen words. It has been shown that Bayesian smoothing using Dirichlet priors outperforms other smoothing methods for concise short queries [303]. Since we construct short queries from the title of profiles, we use this smoothing technique in our retrieval model. Consequently, our retrieval model is a unigram language model with Bayesian smoothing using Dirichlet priors, which ranks the list of matched tweets based on their relevance score. We refer to this relevance score calculated by the above language model as *LM-based relevance score*.

In addition, the weights assigned to semantic groups are used as boosting values for query terms, which indicates the importance of each term in the LM-based relevance score. For instance, if the weight of semantic group s_i is twice the weight of semantic group $s_{i'}$, applying the boosting implies that the appearance of terms from s_i in retrieved tweets is twice as important as the appearance of terms from $s_{i'}$, which is taken into account in the LM-based relevance score calculations. Therefore, the final equation for calculating the LM-based relevance score for tweet t_j is as follows:

$$log\, p(Q|t_j) = \sum_{q_i \in t_j} w_i (log(1 + \frac{tf_{q_i,t_j}}{\mu\, p(q_i|C_d)}) + log\frac{\mu}{|t_j| + \mu}) \tag{4.2}$$

where $p(q_i|C_d)$ is the probability of q_i in the whole set of tweets in day d, and μ is the parameter of the Dirichlet prior. The boosting factor is applied by w_i and the smoothing factor is included in $p(q_i|C_d)$ and μ. The details of how this equation is deduced from the aforementioned unigram language model is explained in [303].

Tweet Relevance Verification

Different from the typical information retrieval scenario, where the user traverses the retrieved list until the information that she needs is found, recommendation and

filtering systems have a special focus on high precision in a short recommendation list. Retrieving irrelevant tweets is disadvantageous, and so every irrelevant tweet that were placed before relevant tweets, it would disturb the user and also deteriorate the performance of the filtering system. In addition, it is reasonable to expect days with no relevant tweets to a particular user interest, where ideally the system should remain silent, as the user should not be disturbed by receiving irrelevant tweets. Such days are referred to as silent days [158]. As a result, tweets that are not sufficiently relevant need to be filtered out, which suggests that a cut-off value needs to be set as a threshold for the minimum acceptable measure of relevance.

For this purpose, we use the weights of semantic groups to calculate a relevance score for each tweet. Let, $t_j = \{t_{j,1}, t_{j,2}, ..., t_{j,|t_j|}\}$ be a retrieved tweet for query Q that is constructed from semantic groups of p, where $t_{j,o}$ is a term in tweet t_j. To verify whether this tweet is relevant to the given profile p, we calculate its relevance score $rel\_score(t_j)$ using Eq. 4.3. The score is the sum of the weights of the semantic groups that appear in the tweet. A semantic group is said to occur in a tweet if at least one term of the semantic group appears in the tweet. If more terms of a same semantic group appear in the tweet, there is no added contribution to the relevance score. If the score is equal to or greater than a predefined threshold θ, tweet t_j is considered relevant. Otherwise, it is deleted from the list of retrieved tweets. In principle, we set $\theta = 1$ in our baseline model and we discuss the sensitivity of this parameter in Section 4.4.6.

$$rel\_score(t_j) = \sum_{s_i \in S} w_i b_i, \quad where \quad b_i = \begin{cases} 1, (s_i \cap t_j) \neq \emptyset \\ \\ 0, (s_i \cap t_j) = \emptyset \end{cases} \tag{4.3}$$

In addition, this technique for filtering out tweets with relevance scores lower than θ is also useful for identifying silent days. If the system cannot find any tweet with relevance score higher than θ, it considers that day to be a silent day.

Novelty Verification

Since novelty is a fundamental feature in information filtering and recommendation systems, we apply Locality Sensitive Hashing [254] to identify clusters of near-duplicate tweets. Only the earliest tweet posted within each cluster is included in the final list of recommended tweets, while the rest are discarded. This strategy is in agreement with the problem definition of avoiding redundant retrieval. In other words, for every pair of retrieved tweets, if the tweet that is published chronologically later does not contain new information compared to the earlier tweet, then the later tweet is redundant with respect to the earlier tweet. Tweets that are not redundant with respect to other tweets already included in the relevant ranked list are considered novel. We define $R = \bigcup_{d' \leq d} R_{d'}$ as the set of all tweets that are selected by the filtering algorithm as relevant to profile p from the start up until the current time in the stream, i.e. R contains selected tweets from the current day d and all previous days. If the relevant tweet t_j is in the same cluster as the relevant tweet $t_{j'}$ that has already been added to R and t_j is published earlier than $t_{j'}$, then $t_{j'}$ is replaced by t_j. Otherwise, t_j is ignored and no changes are applied to the recommended list of tweets.

Ranking Model

Finally, we need to represent the set of selected relevant and non-redundant tweets in a ranked list. We consider the LM-based relevance score of tweets, which is calculated by the retrieval model, as one of the ranking features. Other features are social attributes such as the number of author's followers and followees, the number of tweets published by the author, and the number of tweets the author has liked. Since each of these feature gives a ranked list of tweets, we propose using weighted Borda-Count [1] voting method to combine separate ranked lists into one list.

So far, we described all the steps of our automatic tweet filtering method, where each semantic group has only one member, its corresponding query term. We summarize these steps in Algorithm 3 and we refer to this automatic filtering process as Unsupervised Twitter Information Filtering (UTIF), which is our basis for analyzing different strategies including query expansion.

Algorithm 3 Unsupervised Twitter Information Filtering: UTIF

Initialization step:

for each query term, $s_i = \{q_i\}$ ▷ initialize s_i with query term q_i, extracted from the profile title

for each s_i, initialize $w_i = initWeight(q_i)$ ▷ Subsection "Query Formation"

$Q = \bigcup_{s_i \in S} s_i$ ▷ create query Q with all semantic groups

Filtering step:

$R_d = \emptyset$ ▷ R_d = final ranked list for day d

$l = retrieval(Q)$ ▷ ranked list of tweets returned by the model in Subection "Tweet Retrieval"

while $l \neq \emptyset$ **do**

    $t_j = l.head()$ ▷ remove and get top element

    if $rel\_score(t_j) >= \theta$ **then** ▷ Subsection "Relevance Verification" using Eq. 4.3

        if $\forall t_{j'} \in R, near\_dup(t_j, t_{j'}) == false$ **then** ▷ Subsection "Novelty Verification"

            add t_j to R_d ▷ Add t_j to R if it is not redundant with already retrieved tweets

        else if $near\_dup(t_j, t_{j'}) == true$ **then**

            if t_j is published earlier than $t_{j'}$ **then**

                replace $t_{j'}$ by t_j in R_d

        update $R = \bigcup_{d' \leq d} R_{d'}$ ▷ update R to include newly added tweets

4.3.2 Unsupervised Query Expansion by Semantic Relatedness Methods

Aiming to address the vocabulary mismatch problem, we expand the query by means of semantic relatedness methods. The goal is to find other terms that are semantically related with the terms of the query. Different semantic relatedness models are categorized as distributional [79, 188, 217], lexical knowledge resource-based [214], or hybrid [5] models. From these state-of-the-art methods, we used two methods based on word embedding models, word2vec [188] and GloVe [217], one state-of-the-art model based on Wikipedia link structure [245], which we refer to as WikiSim, and one hybrid method, UMBC Top-N Similarity [103], which uses both WordNet and statistics from a large corpus.

We consider three different strategies for using these models in query expansion. One strategy is that for each query term, we find its most related term to add to its semantic group. We refer to this strategy as *expand_all*, since all query terms are expanded. Word embedding models, word2vec and GloVe, provide a vector representation for each term in the vocabulary. Therefore, we consider cosine similarity between the word vector representations as their similarity score, and the term with the highest similarity score to query term q_i is selected as its most semantically related term. UMBC Top-N Similarity offers a web service [102] that provides a list of most semantically related terms to the given term. Similarly for WikiSim, we obtain the most related Wikipedia concepts for each query term using its web service [244].

Since some terms have multiple meanings, it is difficult to disambiguate their correct meanings independently from their context. Therefore, we consider a second strategy where for each query term, we obtain the top-N most related terms. Terms that are common in at least two sets are added to the query. For instance, after applying the WikiSim model to terms in the query "U.S. Forest Fires", the term "Wildfire" gets added since it appears among the top-N most related terms for both "Fire" and "Forest". This strategy is shown in Eq. 4.4, where $sem(q_i)$ indicates the set of top-N semantically related terms to q_i and $exp(Q)$ is the set of semantically related terms that should be added to the query. In the experiments, we refer to this strategy as *expand_common*.

$$exp(Q) = \bigcup_{i,j=1,i\neq j}^{|\widetilde{Q}|} sem(q_i) \cap sem(q_j) \tag{4.4}$$

Our last strategy for query expansion is similar to expand_all but with the difference that we only expand query terms that are not named entities. The reason for treating named entities differently from other terms is that named entities are usually very specific and are used in interest profiles to narrow the relevant results. Expanding named entities may result in adding terms that are not relevant to user interest. For instance, if a user is interested in events happening in the city of "Rotterdam", the most related word to "Rotterdam", using word2vec [289], is "Amsterdam", whose addition to the query is likely to reduce the precision of the filtering system. We refer to this strategy as *expand_some*. In this case, the semantic groups of named entities, which are not expanded with semantically related terms, have only one member, i.e. the named entity itself.

4.3.3 Active Query Expansion by Semantic Relatedness Methods

Automatic query expansion was described in the previous section. Following our proposed framework in Chapter 1 for using active learning techniques, we propose active strategies for expanding semantic groups and refining their weights, and hence addressing the vocabulary mismatch problem. Our active query expansion is based on finding semantically related term(s) that appear in tweets whose relevance to the profile is confirmed by the information source. The tweets that are selected as labeling requests need to be carefully selected so that we increase the filtering performance while minimizing the number of labeling requests. Before describing these selection strategies, we first explain how queries are expanded and how semantic relatedness methods are used for finding semantically related terms from tweets.

The query expansion is performed as follows: assuming the information source has verified that a tweet t_j is relevant to the given profile, if none of the terms of a semantic group s_i occurs in t_j (we also refer to this situation as s_i is not present in t_j, or $t_j \cap s_i = \emptyset$), either s_i is not an important semantic group for conveying the meaning of the user interest, or it is an important semantic group but the tweet contains semantically related term(s) to the terms in s_i that are yet missing from it.

For instance, let the title of a user interest profile be "FIFA Corruption" with initial semantic groups $S = \{\{FIFA\},\{corruption\}\}$, and the relevant tweet *"SEC launches civil probe into FIFA bribery case: Report http://t.co/8GQUYOHx0q #SEC #FIFA"*, the goal is to infer that "bribery" is a semantically related term to "corruption" for this particular profile and hence be added to its semantic group.

When no semantically related term can be found in the tweet, then we assume that the missing semantic group is considered to be non-important, and therefore its occurrence is not necessarily required for tweet relevance. For instance, if the title of the user interest profile is "Climbing Mount Everest", the term "mount" does not appear in the relevant tweet *"RT @TheFactsBook: Roughly 90 percent of the climbers on Everest are guided clients, many without basic climbing skills"*, since with the occurrence of other semantic groups, the appearance of "mount" is not necessary for projecting the meaning of the user interest. Therefore, it is not an important semantic group for this profile.

Finding Related Terms Using Semantic Relatedness

Now we explain how we find semantically related terms from the verified tweets, e.g. finding "bribery" as a semantically related term to "corruption" from tweet *"SEC launches civil probe into FIFA bribery case: Report http://t.co/8GQUYOHx0q #SEC #FIFA"* for the profile "FIFA Corruption". When none of the terms of s_i are present in the selected tweet t_j, we consider a non-stop term of the tweet t_j, or $t_{j,o}$ where $t_{j,o} = rel\_term(t_j, s_i)$, as the most semantically related term to s_i if the similarity score between them, i.e. $sim\_term(t_{j,o}, s_i)$, is the highest among every possible pairs of terms in tweet t_j and semantic groups in S. This is formulated in Inequalities 4.6 and 4.7. The semantic relatedness score between two terms is calculated by their cosine similarity between their vector representations using word2vec (pre-trained vectors on the Google News dataset). Therefore, the similarity score between a semantic group s_i and term $t_{j,o}$ is the maximum similarity between $t_{j,o}$ and all terms in s_i (See Eq. 4.8). If such a term does not exist, then it means that s_i is not an important semantic group for the profile.

$$rel\_term(t_j, s_i) = \begin{cases} t_{j,o}, & \exists t_{j,o} \in t_j \mid Inequality\ 4.6 \wedge Inequality\ 4.7 \\ Null, & \nexists t_{j,o} \in t_j \mid Inequality\ 4.6 \wedge Inequality\ 4.7 \end{cases} \tag{4.5}$$

$$sim\_term(t_{j,o}, s_i) > sim\_term(t_{j,o'}, s_i) \forall t_{j,o'} \in t_j \tag{4.6}$$

$$sim\_term(t_{j,o}, s_i) > sim\_term(t_{j,o}, s_{i'}) \forall s_{i'} \in S \tag{4.7}$$

$$sim\_term(t_{j,o}, s_i) = \max_{s_{i,r} \in s_i} cos(\mathbf{t_{j,o}}, \mathbf{s_{i,r}}) \tag{4.8}$$

We propose two strategies for selecting the most appropriate tweets to be verified the relevance by the information source, for the purpose of expanding semantic groups. These strategies rank candidate tweets based on two criteria: the likelihood of being relevant to profile p and the likelihood of containing semantically related terms.

Strategy 1: Finding Tweets with Missing Semantic Groups

The first selection strategy is based on finding relevant tweet t_j with missing semantic groups. Therefore, we consider query Q with all of its semantic groups and use our model in Subsection "Tweet Retrieval" to retrieve top recommended tweets. It is reasonable to consider the top ranked tweets as the most probable tweets to be relevant to the given profile p. Therefore, the top tweet that does not contain at least one of the semantic groups is selected as a labeling request.

Let t_j be the selected labeling request that is labeled as relevant by the information source. If by following the steps in Subsection "Finding Relevant Terms Using Semantic Relatedness", semantically related term(s) to the missing semantic group(s) are found, we add them to their corresponding semantic groups. Otherwise, we consider the missing semantic group(s) to be less important and we adjust their weights accordingly. More specifically, we distribute the weight of the missing semantic groups among the other occurring groups. The intuition behind this adjustment is to increase the importance of the semantic groups that do occur in the tweet. There may be several relevant tweets missing the same non-important semantic group. However, each

non-important semantic group can be used in modifying the weights of other semantic groups only once. For this purpose, we define a set $\hat{S}$ to contain non-important semantic groups. This set is initialized to an empty set before applying any selection strategy. The steps related to this strategy are summarized in Algorithm 4 and also shown in Fig. 4.2 (left hand side).

Algorithm 4 Strategy 1: Finding Tweets with Missing Semantic Groups

```
1:  Q = ⋃_{s_i∈S} s_i                          ▷ create query Q with all terms in its semantic groups, (Fig 4.2 a.1)
2:  l = retrieval(Q)                           ▷ ranked list of tweets returned by "Tweet Retrieval", (Fig 4.2 a.2)
3:  while l ≠ ∅ do
4:      t_j = l.head()                         ▷ remove and get top element, (Fig 4.2 a.3)
5:      if ∃s_i ∈ Q, that s_i ∩ t_j = ∅ then   ▷ (Fig 4.2 a.4)
6:          lbl = label(t_j)                   ▷ ask the information source to label t_j
7:          if lbl == relevant then            ▷ if tweet t_j is relevant to profile p, (Fig 4.2 c.1)
8:              set sum_nonpresent = 0         ▷ keeps the weights of missing semantic groups
9:              for (s_i ∩ t_j = ∅) ∧ (s_i ∉ E) do   ▷ lines 8-17 are shown in Fig 4.2 c.2
10:                 t_{j,o} = rel_term(t_j, s_i)   ▷ see "Finding Relevant Terms Using Semantic Relatedness"
11:                 if t_{j,o} ≠ Null then
12:                     s_i = s_i ∪ t_{j,o}        ▷ t_{j,o} is a semantically related term to terms in s_i
13:                 else if s_i ∉ Ŝ then
14:                     sum_nonpresent+ = w_i
15:                     Ŝ = Ŝ ∪ s_i                ▷ s_i is not an important semantic group
16:             for s_i ∩ t_j ≠ ∅ do               ▷ increase weight of s_i, which is present in t_j
17:                 w_i+ = sum_nonpresent / (|S|−|Ŝ|)
```

Strategy 2: Excluding Semantic Groups from Query

In this second strategy, we expand the query by intentionally excluding semantic groups one by one and then trying to find related terms to the excluded semantic groups from relevant tweets. Let us assume that we want to expand semantic group s_i. We first form a query containing all the other semantic groups except s_i, and find the matched tweets to this query that do not contain any term from the excluded semantic

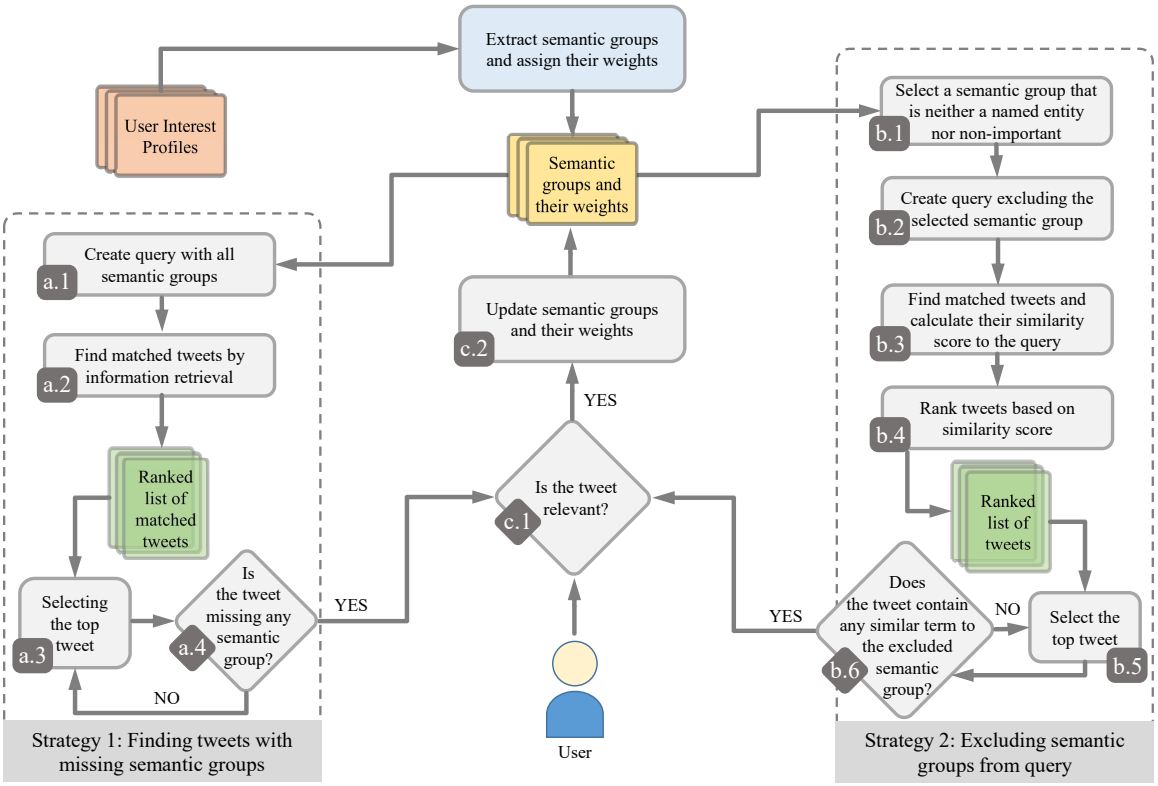


Figure 4.2: Proposed active query expansion strategies for tweet filtering based on user interest profiles. These strategies are applied in an interleaved fashion until the information source is satisfied with the performance of the filtering or a predefined limit is reached. Strategy 1 is shown on the left hand side, where steps a.1, a.2, a.3, a.4 are equivalent to lines 1, 2, 4, 5 of Algorithm 4, respectively. Strategy 2 is shown on the right hand side, where steps b.1-b.6 are equivalent to lines 1-7 of Algorithm 5. c-1 and c-2 include different steps of updating semantic groups in both strategies, i.e. lines 8-17 in Algorithm 4, and line 10 in Algorithm 5.

group[1]. Then, to select the tweet with the highest probability of containing related terms, a semantic relatedness score is calculated for each matched tweet indicating its similarity to the set of semantic groups, S. Tweets with high scores are more likely to contain related terms for expanding s_i. The semantic relatedness score between each matched tweet and S is calculated as follows:

$$sim\_tweet(t_j, S) = \frac{1}{|S|} \sum_{s_i \in S} \max_{1 \leq o \leq |t_j|} sim\_term(t_{j,o}, s_i) \tag{4.9}$$

where $sim\_term(t_{j,o}, s_i)$ is calculated based on Eq. 4.8 and $|S|$ is equal to the number of semantic groups, which is also equal to $|\widetilde{Q}|$ as for each query term $q_i \in \widetilde{Q}$ there is a semantic group s_i.

After calculating the semantic relatedness for all matched tweets, we sort them based on this value. Let us assume that tweet t_j has the highest score. Therefore, t_j has the highest probability of containing a semantically related term to the terms in the excluded semantic group (missing group). We consider this tweet as a potential labeling request and follow the steps in Subsection "Finding Relevant Terms Using Semantic Relatedness" to find the term $t_{j,o} \in t_j$ as the potential related term to the terms in s_i. If this term exists, then we select tweet t_j as a labeling request. If the information source labels this tweet as relevant, then we expand s_i by adding $t_{j,o}$ as its semantically related term. Otherwise, the next tweet in the ranked list is selected as a potential labeling request. These steps are summarized in Algorithm 5 and illustrated in Fig. 4.2 (right hand side). It is important to note that we apply this strategy only to semantic groups that do not contain any named entity and are not identified as non-important in the first strategy.

Active Tweet Filtering: ACTIF

Now that all the steps of the proposed method are explained separately, we integrate them as our ACtive Twitter Information Filtering (ACTIF) algorithm. Algorithm 6 summarizes the complete ACTIF steps. This algorithm contains the proposed selection strategies already described and how the system learns from the provided labels as well as our method for finding and ranking relevant tweets. We continue

[1]This is executed by performing a boolean query in such a way that the occurrence condition of the excluded semantic group is set to MUST_NOT and the other semantic groups to SHOULD.

Algorithm 5 Strategy 2: Excluding Semantic Groups from Query

for all $s_i \in S$ that $(s_i \cap E = \emptyset) \wedge (s_i \notin \hat{S})$ **do** ▷ Fig. 4.2, b.1
  $Q = \bigcup_{s_{i'} \in S, i' \neq i} s_{i'}$ ▷ Q includes all terms in semantic groups except terms in s_i, (Fig. 4.2, b.2)
  $l = sortBySimilarity(Q, S)$ ▷ sort tweets by relatedness to S, see Eq. 4.9, (Fig. 4.2, b.3-b.4)
  while $l \neq \emptyset$ **do**
    $t_j = l.head()$ ▷ remove and get top tweet, (Fig. 4.2, b.5)
    $t_{j,o} = rel\_term(t_j, s_i)$ ▷ Subsection "Finding Relevant Terms Using Semantic Relatedness"
    if $t_{j,o} \neq Null$ **then** ▷ If $t_{j,o}$ exists (Fig. 4.2, b.6)
      $lbl = label(t_j)$ ▷ ask the information source to label t_j
      if $lbl == relevant$ **then** ▷ if tweet t_j is relevant to profile p, (Fig. 4.2, c.1)
        $s_i = s_i \cup t_{j,o}$ ▷ add term $t_{j,o}$ to semantic group s_i, (Fig. 4.2, c.2)

applying strategies 1 and 2 until the user is satisfied with the performance of the filtering or some limit is reached. After that, having the expanded semantic groups and their adjusted weights, we follow the filtering step of Algorithm 3 for finding relevant tweets.

Algorithm 6 Active Twitter Infromation Filtering: ACTIF

Initialization step
$no\_labeling\_requests = 0$
while $no\_labeling\_requests < limit$ **do**
  perform Strategy 1
  perform Strategy 2
$Q = \bigcup_{s_i \in S} s_i$ ▷ create a query including all query terms from all semantic groups
Filtering step ▷ See Algorithm 3

4.4 Evaluation

We evaluate our proposed methods for tweet filtering, UTIF and ACTIF, in separate set of experiments. We first discuss the experiments related to the automatic query expansion and then the results of the proposed active strategies, ACTIF, and its comparison with SVMRank, a supervised learning-to-rank method. All the proposed methods are implemented using Java and Apache Lucene [76].

4.4.1 Dataset

The Twitter dataset used in this study was collected during TRECMicroB evaluation (10 days). Our system listened to tweets on the Twitter's streaming API [272, 157] and since we likely obtained the same tweets as other participants [209], we can compare our system against those in the challenge. We gathered 40,264,332 tweets during this period , but considered only English tweets (16,302,498 tweets) in our experiments. Although all of the experiments in this chapter are from post hoc runs on the gathered dataset, we assume there is a stream of tweets. Therefore, when filtering tweets for a specific day, we can use the knowledge learned from earlier days, but not the knowledge from the days that follow.

The set of user profiles is also provided by TRECMicroB organizers and is publicly available [267]. In addition, the relevance score for a selected subset of tweets is available from the TREC website [266]. This subset is selected from the submitted tweets by all participating systems in the competition. Each tweet is judged by assessors independently with respect to the user's interest profile. The set of user profiles contains 225 profiles and the participating systems were supposed to find relevant tweets for all of these profiles. However, only 51 profiles were selected for the evaluation phase of TRECMicroB, and only tweets submitted for these 51 profiles were judged. Based on the data reported in the overview of the competition, four profiles have zero relevant tweets, while the remaining 47 have three or more relevant tweets [158]. We consider the evaluation subset of profiles in our experiments. In addition, if our system returns a tweet which has not been judged, it is treated as an irrelevant tweet. Considering all non-judged tweets as irrelevant is common in similar research studies and also recommended by TRECMicroB organizers.

4.4.2 Evaluation Metrics

One of the well-known metrics for evaluating ranked lists is nDCG@k [251, 129]. This metric, which is also used in TRECMicroB, takes into account both the relevance of the recommended item and its position in the ranked list, and is defined as follows:

Definition 8 *normalized Discounted Cumulative Gain:* $nDCG@k = \frac{DCG@k}{IDCG@k}$, *where* $DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{log_2(i+1)}$, i *indicates the position in the ranked list and* rel_i *is the*

true relevance score of the tweet at position i. IDCG@k is the ideal DCG@k value and is achieved by ranking the list of relevant tweets by their actual relevance score.

The nDCG value is calculated over the top k retrieved results for each profile per day. Therefore, the nDCG value for each profile is the average of its value over the evaluation period, and the final value of this metric is its average over all profiles in the evaluation set (51 profiles). We set k equal to 10, which is consistent with the TRECMicroB, and 5. In addition, we consider the Mean Average Precision (MAP) [180] over all retrieved results, i.e. the top 100 tweets as defined in Section 4.1.1, as another evaluation metric.

Based on the above definition, nDCG@k can be calculated where there is at least one relevant document in the dataset, or in our case, one relevant tweet in a specific day. However, as we discussed in Subsection "Tweet Relevance Verification", there are silent days with no relevant tweets for particular profiles. One strategy for evaluating silent days is to assign the perfect score to the system if it correctly identifies silent days [158]. In other words, the system gains the perfect score for that day (nDCG = 1), otherwise it is penalized to its minimum value (nDCG = 0). It has been shown that this strategy is very sensitive to silent days and it can favor systems that only retrieve a few tweets and are silent most of the time [262]. Out of 51 test topics in this dataset, there are 28 topics with no relevant tweets for at least one day in the evaluation period. Therefore, a system which does not recommend anything will still have an nDCG value above zero.

In order to have a better evaluation of the system, Tan et al. [262] proposed two versions of nDCG: nDCG-1, which is the same as the metric explained above, and nDCG-0, which assigns the same score to all systems on silent days (nDCG-0=0) regardless of the retrieved results. These two metrics are also used as evaluation metrics for the 2016 TREC Real-Time Summarization Track [268]. We use both metrics in the experiments. Similarly, for MAP, we consider two metrics: MAP-1, which considers the same strategy as nDCG-1 for silent days and MAP-0, where systems receive the same score MAP-0=0 regardless of the retrieved results.

4.4.3 Threshold Removal for Non-Silent Days

As discussed in Subsection "Tweet Relevance Verification", we considered that recommending irrelevant tweets deteriorates the performance of tweet filtering systems. That is the reason that UTIF (Algorithm 3) calculates a relevance score for each matched tweet and compares it against the threshold θ, so that only a subset of the retrieved tweets are regarded as relevant. However, based on the evaluation metrics nDCG-1@k and nDCG-0@k, if a tweet filtering system found all relevant tweets and ranked them correctly, it would achieve the perfect nDCG score, regardless of the number of irrelevant tweets that are placed at the end of the recommended list, i.e. after position k or after the last relevant tweet if the number of relevant tweets are less than k. On the other hand, every irrelevant tweet that were placed before relevant tweets, it would deteriorate the nDCG score.

The language modeling approach used for tweet retrieval returns a ranked list of tweets based on their relevance to the query. Therefore, considering nDCG metrics, there is no need for verifying the relevance of retrieved tweets. However, days when the system should remain silent are still a challenge as the system should not retrieve any tweet to avoid affecting its performance negatively. Therefore, we still need to verify the relevance of tweets, but only for identifying silent days. In other words, we apply the relevance verification step until we find at least one tweet with a score equal to or greater than θ in a day. After that, we ignore the tweet relevance verification step for all the other retrieved tweets by the retrieval model for that day.

In addition, in the active retrieval scenario, we have some labeled tweets for each day, which can be used for the better identification of silent days. Therefore, the filtering system remains silent if either one of the following two conditions are met: 1) none of the labeling requests for that day are labeled as relevant, 2) there are no tweets in the retrieved list with the relevance score greater than or equal to θ. We refer to this modification of ACTIF as "ACTIF-Verify" since the relevance verification step is not applied to all of the retrieved tweets. The results of this method is reported in Subsection 4.4.5.

4.4.4 Results: UTIF and Automatic Query Expansion

The performance of UTIF and a subset of discussed strategy combinations of weight assignment to semantic groups and automatic query expansions are presented in Table 4.2. The results show that the identification of named entities in user profiles leads to an improvement of the nDCG and MAP-1 metrics. None of the query expansion strategies managed to improve on the previous results. The closest scores using query expansion were obtained when jointly used with NEI, which would indicate that adding query expansion does not overcome the benefits of weighting named entities differently. In addition, in all different combinations, *expand_some* outperforms *expand_all* except when using UMBC, which does not contain most of the named entities in its pre-trained similarities and therefore does not return any semantically related terms for them. In this case, results for expand_some and expand_all are the same. It was explained in Subsection 4.3.2 that expand_common first extracts for each query term a list of its top-N most related terms and then add terms that are common in at least two lists to the query. In the experiments, size of the semantically related list for each query is set to 10, i.e. $N = 10$.

Table 4.2: Results when combining UTIF with different strategies (all = expand_all, som = expand_some, com = expand_common) using nDCG@10,@5 and MAP-1

Method	expansion			NEI	nDCG-1		nDCG-0		
	all	som	com		@10	@5	@10	@5	MAP-1
UTIF					0.3312	0.3380	0.1057	0.1125	0.2865
				✓	**0.3651**	**0.3701**	0.1435	0.1486	**0.3101**
UTIF+WikiSim			✓	✓	0.3595	0.3646	**0.1470**	**0.1508**	0.3029
		✓		✓	0.3433	0.3446	0.1413	0.1426	0.2906
	✓			✓	0.3255	0.3252	0.1255	0.1252	0.2732
		✓			0.3290	0.3360	0.1075	0.1145	0.2827
	✓				0.3062	0.3100	0.1100	0.1062	0.2629
UTIF+word2vec			✓	✓	0.3555	0.3599	0.1437	0.1479	0.2997
		✓		✓	0.3557	0.3602	0.1419	0.1465	0.3000
	✓			✓	0.3255	0.3252	0.1255	0.1252	0.2732
		✓			0.3290	0.3360	0.1075	0.1145	0.2827
	✓				0.3127	0.3183	0.1009	0.1065	0.2675
UTIF+GloVe			✓	✓	0.2874	0.2906	0.1187	0.1220	0.2445
		✓		✓	0.3400	0.3413	0.1322	0.1334	0.2914
	✓			✓	0.2808	0.2765	0.1023	0.0981	0.2448
		✓			0.3168	0.3187	0.0991	0.1011	0.2772
	✓				0.2975	0.2989	0.0936	0.0949	0.2597
UTIF+UMBC			✓	✓	0.3274	0.3265	0.1313	0.1304	0.2802
		✓		✓	0.3243	0.3228	0.1302	0.1286	0.2767
	✓			✓	0.3243	0.3226	0.1301	0.1285	0.2765
		✓			0.2996	0.2995	0.0976	0.0975	0.2604
	✓				0.2996	0.2993	0.0976	0.0974	0.2603

Moreover, it is possible to train word2vec vector representation of words on other datasets. We trained word2vec on our Twitter dataset considering three different values for the dimensionality of word vectors representations: 100, 300, and 500, and used the trained vectors for query expansion. The results were not significantly different from the results when the pre-trained vectors on the Google News dataset are used. In addition, word2vec performs better compared to GloVe and UMBC, while the best results are achieved when using WikiSim semantic relatedness model.

Let us assume that the title of a user's interest profile is "Mr. Holmes Movie". A subset of returned tweets by UTIF when combined with NEI and expanding the query using word2vec embeddings is shown in Table 4.3. In all cases, we set $\theta = 1$. The baseline form of UTIF, considering equal importance for all query terms, misses relevant tweets, while expand_all retrieves some irrelevant tweets. For instance, expanding "Holmes" results in adding terms such as "Richards" to the query which in turn attracts irrelevant tweets. This shows that while assigning a higher importance to named entities results in finding more relevant tweets, including them in query expansion adds tweets that are not relevant to the profile.

Table 4.3: Sample tweets returned/missed by UTIF and its combinations for profile "Mr. Holmes Movie" considering word2vec embeddings in query expansion and $\theta = 1$

Tweet text	Score	Recommended by			
		UTIF	NEI[a]	all[b]	some[c]
Movie Review: 'Mr. Holmes' Is A Warm Tribute To The Elderly Great Detective http://t.co/m6IgG37qSO	Highly Relevant	✓	✓	✓	✓
Sherlock Holmes, Family Man - The new Mr. Holmes film does what could not have been predicted and does it wonderfu... http://t.co/jtBDXRTRFh	Highly Relevant		✓	✓	✓
Saw "Mr. Holmes," a passable fictional account of what if Sherlock Holmes were a real old man fictionally fictionalized. 2.5 of 4 stars	Somewhat Relevant		✓	✓	✓
#Epic ADAM RICHARDS EPIC FILM REEL PRODUCED AND DIRECTE: https://t.co/7r3aktcaGi #MultiGenreGenius	Not Relevant			✓	

[a] UTIF+NEI

[b] UTIF+NEI with query expansion (expand_all)

[c] UTIF+NEI with query expansion (expand_some)

The results of our method along with the top 4 teams that participated in TRECMicroB are presented in Table 4.4. There are three different types of system categories based on the amount of human involvement in the recommendation task: automatic

(no human input is allowed), manual preparation (human input is allowed only before the evaluation starts), and manual intervention (human input is allowed all the time). UTIF and its combinations discussed in this section fall under the automatic category. The results show that our unsupervised approach based on the language model-based retrieval combined with the assignment of higher weights to named entities outperforms all automatic and manual methods except one method, SNACS LB(NUDTSNA) [310], where they use a boolean logic keyword filter for determining words that need to be included in retrieved tweets to be relevant, and words that are unnecessary but could increase the relevance score of tweets. To identify the necessary from unnecessary words from profiles, they computed the tf-idf values of the words and checked them against two thresholds. These thresholds are set based on empirical settings. To illustrate the importance of the evaluation metric itself, we also added the results of a "NULL" run returning an empty list of tweets for all profiles in all days as a baseline. Results from [158] show that out of 42 submitted runs to this track, only 4 teams outperformed this baseline.

Table 4.4: Results of UTIF+NEI and top 4 TRECMicroB participants sorted by nDCG@10

Run(Group)	nDCG@10	Type
SNACS LB(NUDTSNA)	0.3670	automatic (empirical settings)
UTIF+NEI	**0.3651**	**automatic**
SNACS(NUDTSNA)	0.3345	automatic
CLIP-B-0.6(CLIP)	0.2491	automatic
umd hcil run03(umd hcil)	0.2471	automatic
NULL	0.2470	automatic

4.4.5 Results: ACTIF and Active Query Expansion

In order to better evaluate the effectiveness of the proposed active strategies, ACTIF is compared against two baselines: SVMRank [131], which is a supervised learning-to-rank method, and Round-Robin. We first explain these methods and then discuss their performance in comparison with ACTIF.

Supervised Learning-to-Rank Method: SVMRank

As a supervised learning algorithm, SVMRank requires labeled data to learn how documents should be ranked. Since we assume no labeled data is available, we need to sample tweets to be labeled by an information source and then use them to train SVMRank. However, the large amount of tweets published every day makes it difficult to find relevant tweets for training, since only a small fraction of tweets are relevant to the profile. The percentage of relevant tweets in the dataset for the five profiles with the highest number of relevant tweets is reported in Table 4.5. We can conclude that the probability of finding relevant tweets by random selection is low.

Table 4.5: Percentage of relevant tweets in the collected dataset, in the union of all matched tweets, and in the top two ranked tweets by the language model

Profile	Sampling Population		
	dataset	matched tweets	LM-top2
MB401	0.0125%	0.5036%	70.00%
MB344	0.0104%	8.6545%	95.00%
MB243	0.0029%	2.4818%	45.00%
MB236	0.0028%	2.5982%	80.00%
MB246	0.0026%	1.6548%	50.00%

Consequently, we formulate a query from the terms in the title of each profile and consider the union of all matched tweets as the sampling population for that profile. This makes the sampling population smaller and increases the chance of finding relevant tweets. This is shown in Table 4.5 under column "matched-tweets". Furthermore, column "LM-Top2" shows the percentage of relevant tweets in the top two tweets by using a retrieval based on our unigram language model explained in Subsection "Tweet Retrieval". The values reported in this table are calculated considering all 10 days in the evaluation period.

We further evaluate different sampling strategies for SVMRank by considering its performance in filtering tweets. In all sampling strategies, we consider two tweets for each profile per day. Considering more tweets is equal to increasing user effort in real settings, where a human user is involved. From the different options for sampling, we considered the following strategies:

- Rand2: random selection of two tweets from the set of matched tweets to the

Table 4.6: SVMRank results with different sampling strategies considering 1020 labeling requests

Sampling Method	nDCG-1@10	nDCG-0@10
LMTop2+Neg	0.4539	0.1911
LMTop2	0.4493	0.1865
LMTop2-NV	0.4341	0.1772
LMTop1End1	0.4063	0.1455
Rand2	0.2766	0.0276

query.

- LMTop1End1: selecting one tweet from the top of the ranked list returned by our retrieval method and one tweet from the end of the list. The reason for this strategy is that SVMRank learns how to rank tweets by learning from the order of the training samples. Therefore, one tweet from the top and one from the bottom of the list are reasonable choices for allowing diversity in its training.
- LMTop2: selecting the top two tweets from the ranked list returned by our retrieval method.
- LMTop2+Neg: similar to LMTop2 with regard to the labeling requests. However, if there are no irrelevant tweets among all the labeled tweets (including previous days), the last tweet in the ranked list is added as a not relevant tweet to the training set without asking the information source. As the last matched tweet has the lowest probability of being relevant, we save a labeling request by assuming it is not relevant.
- LMTop2-ND: similar to LMTop2, but without considering novelty verification in selecting labeling requests.

Labeled tweets from previous days are used in the training data for the following days. Therefore, the training data for the last day in the evaluation period contains 20 labeled tweets. Since there are 51 profiles in our dataset, in total 1020 labeling requests are asked from the information source. The feature vectors for training and test instances are tf-idf values of the text of the tweets after URLs are removed.

The results of this evaluation are reported in Table 4.6. In all of the sampling strategies, except LMTop2-NV, near-duplicate tweets are identified and removed in order to prevent redundant labeling requests. LMTop2-NV is the same as LMTop2,

but when novelty verification is not applied. The difference between the performance of these two methods confirms that preserving the labeling requests by removing near-duplicate tweets improves the performance of SVMRank. LMTop2+Neg outperforms other strategies and is considered as the sampling strategy for constructing training data of SVMRank in the rest of the experiments. If no relevant tweet exists in the training data of a day (excluding the tweets from previous days), that day is considered as a silent day and the system remains silent. The steps of applying SVMRank for filtering tweets are summarized in Appendix H.

Similar to our discussion in Section 4.4.3, we expect that if SVMRank learns properly, it will place the irrelevant tweets towards the end of the ranked list. For the purpose of evaluation, we apply the relevance verification step to the results of SVMRank. In other words, we use Eq. 4.3 to calculate the relevance score of each tweet and if it is not equal to or higher than θ, then it is removed from the ranked list. However, the order of the remaining tweets is not changed from what was suggested by SVMRank. We refer to this method as SVMRank+Verify, since the relevance verification step is added to SVMRank.

Round-Robin

RoundRobin is based on our unsupervised filtering system UTIF, but also alters the results by using labeling requests to filter out irrelevant tweets and sort relevant tweets based on their actual relevance score. In other words, the system does not learn from labeled tweets and only integrates them in the final ranking. Therefore, the labeling requests are selected from the results, i.e. ranked tweets returned by UTIF algorithm. The first ranked tweet by an information filtering system has the highest contribution to its nDCG value, and as the position of tweets become lower in the list, their contribution become smaller. Therefore, it is reasonable to select the labeling requests from the top of the ranked list to gain most of the provided labels.

For this method, UTIF is applied first, and then top ranked tweets are selected as labeling requests. After the information source labeled the selected tweets, tweets labeled as irrelevant are removed from the suggested ranked list, and tweets that are labeled as relevant are sorted based on their relevance level (highly relevant or somewhat relevant) and placed at the beginning of the ranked list. This is because

these tweets have been manually labeled as relevant, and thus should be placed higher than tweets estimated to be relevant by the algorithm. These steps are summarized in Appendix I.

We refer to this baseline as RoundRobin because for multiple profiles, we first select the top ranked tweet for all the profiles and then continue with the next ranked tweets in a circular order until the number of labeling requests reaches its limit, or the user stops labeling (when human users are involved). This approach of selecting instances from multiple ranked lists is called Round-Robin. Results of this method is compared with the proposed method and SVMRank in the next section.

Wilcoxon Signed Rank Test

The Wilcoxon signed rank test is used to determine statistically significant differences between ACTIF and the baselines: SVMRank and Round-Robin. It is a non-parametric statistical test, which is suitable for comparing matched groups where the difference between the values of the target variable can be ranked, but the magnitude is not important. The settings of our experiments are compatible with these assumptions. The summary of the methods are reported in Table 4.7 and the results are shown in Table 4.8. We associate a digit to each method and use them in columns "SS-1", "SS-0", and "SS-MAP0" to report the methods that are statistically outperformed by the "Target Method" considering nDCG-1@10, nDCG-0@10 and MAP-0, respectively. For instance, ACTIF outperforms UTIF+NEI considering all metrics. "*" is used for $p \leq 0.05$ and "**" is used for $p \leq 0.01$. It should be noted that these p-values are computed in a pairwise manner. In addition, the experiments in this section consider NEI for all methods.

The results of SVMRank+Verify method indicate that the performance of SVM-Rank can be improved by verifying the relevance of the retrieved tweets. More detailed performance of UTIF, ACTIF and SVMRank for each individual profile is reported in Appendix J.

In order to have a better understanding of the performance of the methods, we vary the number of labeling requests, from 102 to 2040 in 20 steps. nDCG@10 values for ACTIF, ACTIF-Verify, SVMRank, and Round-Robin are shown in Fig. 4.3. We observe that ACTIF outperforms SVMRank and improves nDCG-0 significantly.

Table 4.7: Summary of different methods. Columns from left to right represent: the name of the method, the method for selecting labeling requests, whether the relevance verification step is applied to all tweets or only for the identification of silent days, and the method for ranking tweets.

Method	Labeling Requests	Relevance Verification	Ranking Method
ACTIF	actively selected	for all tweets	LM-Based
ACTIF-Verify	actively selected	only to identify silent days	LM-Based
SVMRank	LMTop2+Neg	not applied	SMVRank
SVMRank+Verify	LMTop2+Neg	for all tweets	SVMRank
Round-Robin	top tweets	for all tweets	LM-Based
UTIF+NEI	not applicable	for all tweets	LM-Based

Table 4.8: nDCG-1@10, nDCG-0@10, and MAP-0 for ACTIF, SVMRank, UTIF+NEI and Round-Robin. SS-1, SS-0, and SS-MAP0 indicate methods (using their associated digits) that are statistically outperformed by the "Target Method" using the Wilcoxon signed rank test. All active methods consider 1020 labeling requests.

Target Method	nDCG1	SS-1	nDCG0	SS-0	MAP-0	SS-MAP0
1-ACTIF	0.4524	6**	0.2407	3** 4** 5** 6**	0.1614	3** 4* 5** 6**
2-ACTIF-Verify	0.4659	6**	0.2365	3** 4* 5** 6**	0.1669	3** 4* 5** 6**
3-SVMRank	0.4539	6**	0.1912	6**	0.1328	6*
4-SVMRank+Verify	0.4632	5* 6**	0.2043	6**	0.1374	6**
5-Round-Robin	0.4154	6**	0.1684	6**	0.1113	6**
6-UTIF+NEI	0.3651	-	0.1435	-	0.0885	-

Moreover, our proposed method has higher nDCG-1 for labeling requests less than 1020, which is desirable given the inherent cost for labeling. The difference between Round-Robin method and other methods, ACTIF and SVMRank, highlights how much these systems learn from the provided labels.

It is worth highlighting that ACTIF performed significantly better for nDCG-0. This shows the superiority of our approach when silent days are not considered in the comparison. When considering nDCG-1, which has a drastic effect on silent days, our approaches seem to be on pair with the SVMRank variants for a relatively high number of labeling requests (above 900 labeling requests), but performing better when little labeling is provided. All of the active-based approaches significantly overcome the unsupervised filtering.

In addition, in Section 4.2, we discussed two recent studies by Tan et al. [261] and Zhu et al. [309] with similar problem definitions. Tan et al. considered 10 labeling requests for each profile per day for adjusting a dynamic threshold. Taking into

account 51 profiles in the whole dataset and 10 days in the evaluation period results in a much higher number of labeling requests than what our proposed strategies considered in the experiments. Moreover, Zhu et al. reported the nDCG-1@10 value of their systems in comparison with top TRECMicroB participants. The best result of their system achieved 0.367 for nDCG-1@10, which is slightly higher than the results of our UTIF method (nDCG-1@10 = 0.3651). It is also important to note that no labeled data are used in UTIF, while Zhu et al. used labeled data for the empirical settings of the parameters of their method. They also calculated a manufactured performance, which is the average of the best results obtained for each profile from applying different algorithms, including top 4 systems in TREC 2015. They refer to this value as an upper bound for TREC 2015 Microblog dataset since it is not the results of a single automatic or manual method, but the average of the best results reported for this problem. Comparing the value of this upper bound (nDCG-1@10 = 0.5014) with the results of ACTIF indicates the effectiveness of our approach and confirms that with incorporating user knowledge in the filtering process and employing active strategies, the performance of the systems can be improved significantly with limited user supervision.

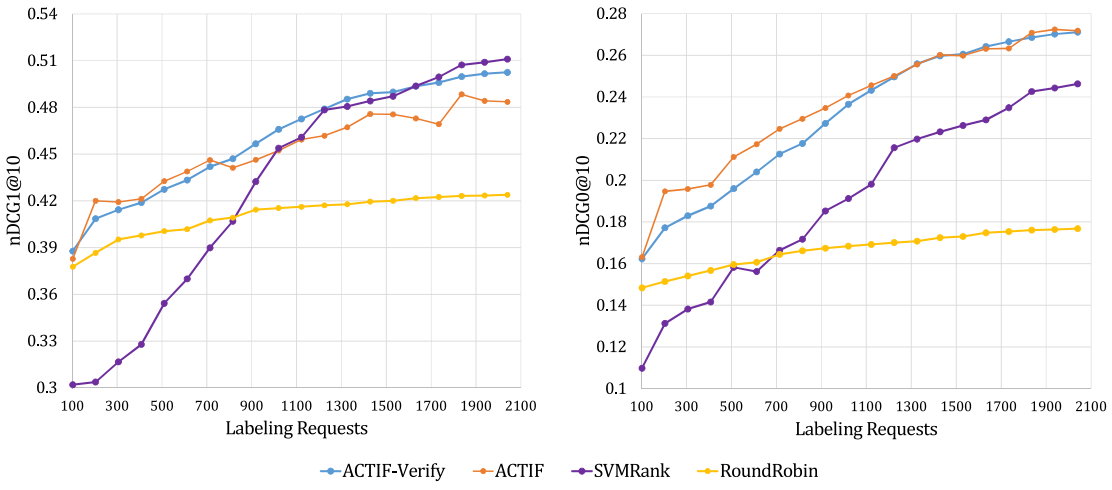


Figure 4.3: nDCG-1@10 and nDCG-0@10 of ACTIF, SVMRank, and Round-Robin for different number of labeling requests.

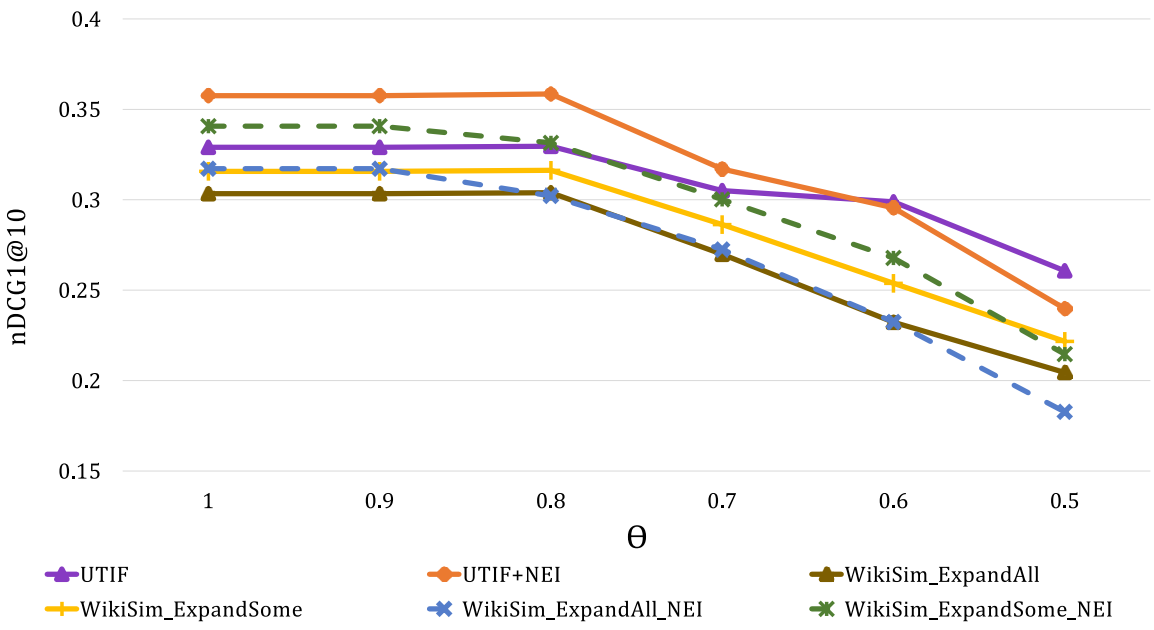


Figure 4.4: Results of UTIF and its combinations with WikiSim using nDCG-1@10 for different values of θ.

4.4.6 Discussion

In Subsection "Tweet Relevance Verification", we considered a threshold $\theta = 1$ to filter out irrelevant tweets. To evaluate the sensitivity of results to this threshold, we vary its value in the range [0.5,1.0] in 5 steps. The nDCG-1@10 values for our unsupervised tweet filtering, UTIF, and its combination with automatic query expansion using WikiSim is shown in Fig. 4.4. As the value of this threshold is reduced, the system considers more tweets as being relevant and the chance of recommending irrelevant tweets gets higher, which in turn reduces nDCG scores. It is worth mentioning that each profile title in our dataset has 3.2 query terms in average (after removing stop words). Therefore, based on our strategies for assigning weights to each semantic group (see Subsection "Query Formation"), the weight of each semantic group is around 0.3 on average, which indicates that reducing θ to values greater than or equal to 0.8 does not have a significant impact on the nDCG value.

We also investigated the semantic relatedness methods besides the retrieval framework of TIF. Inspired by [142], where they calculate text similarity from the combination of word semantic relatedness, we use the word embedding models word2vec and GloVe for calculating the similarity for each tweet-profile pair. Therefore, for each

profile, tweets are ranked based on their similarity to that profile. For calculating the similarity for each tweet-profile pair, we considered two approaches: 1) the similarity between the average vectors of the composing terms of the title of the profile and the tweet, and 2) the average of the maximum pairwise similarities between terms of the title of the profile and the tweet. The first strategy is shown in Eq. 4.10 where $\mathbf{q_i}$ and $\mathbf{t_{j,o}}$ are the word vector representations of terms q_i and $t_{j,o}$, and $\widetilde{Q}$ is the set of terms extracted from the title of each profile. The latter strategy uses Eq. 4.9 for calculating semantic similarity between tweets and profiles.

$$score\_avg(t_j, \widetilde{Q}) = (\sum_{q_i \in \widetilde{Q}} \frac{\mathbf{q_i}}{|\widetilde{Q}|}) \cdot (\sum_{t_{j,o} \in t_j} \frac{\mathbf{t_{j,o}}}{|t_j|}) \tag{4.10}$$

We again considered different thresholds as cut-off points of the similarity scores in order to prevent recommending tweets with low similarity values, and to identify silent days as well. Based on the results in Table 4.9, neither of these approaches outperformed the NULL method, which is silent for all profiles in all days. This is consistent with the results of our participation in TRECMicroB, where we used Wikipedia concepts and Google tri-grams to calculate a semantic similarity value between a particular tweet and all profiles and assigned the tweet to the profile with the highest similarity value [52]. In addition, these approaches are CPU and memory intensive, since it calculates the similarity between all tweets and all profiles in the dataset.

Table 4.9: Results of the semantic text relatedness methods based on word embeddings using nDCG-1@10

Threshold	Similarity by Average		Similarity by Maximum	
	word2vec	GloVe	word2vec	GloVe
0	0.0670	0.0320	0.0088	0.0068
0.6	0.1721	0.1040	0.1576	0.0967
0.7	0.2210	0.1169	0.2176	0.1274
0.8	0.2458	0.1309	0.2373	0.2333

As a final note, we discussed the incorporation of social features in the final ranking of relevant tweets in Subsection "Ranking Model". We used different features such as the number of authors' followers, followees, and published tweets, and observed that including these features does not improve the performance of our filtering system

significantly. In addition, among the features we examined, the number of authors' followees resulted the best performance. Therefore, it is considered in all different methods that are used in the experiments of this chapter.

4.5 Conclusions

We presented a framework for tweet filtering task and we used this framework to analyze different strategies for query expansion and verification of the relevance of the tweet. With respect to our methods for unsupervised tweet filtering, results demonstrated that named entities have an important role in the precise recommendation of tweets based on user interest profiles. UTIF+NEI performs the best compared to all TRECMicroB participants (in the automatic category). Our analysis also showed that incorporating semantic relatedness methods does not seem to improve tweet filtering—at least in the TRECMicroB dataset—either when used for automatic query expansion or when used for calculating the semantic relatedness between tweets and profiles.

Observing that automatic query expansion does not improve the results motivated us to involve an information source in the process to actively expand queries. Our active strategies modify the query using semantic relatedness methods and two selection strategies, which are based on the hypothesis that if none of the terms in a semantic group appear in a relevant tweet to a particular profile, it is because either another semantically related term to these terms has occurred in the tweet or the semantic group is not important for that profile. To find related terms to expand the query with, we used pre-trained word2vec vectors to identify related terms to the missing semantic groups from relevant tweets. The experiments supported our hypothesis and demonstrated the effectiveness of the selection strategies. The proposed method was also compared against SVMRank and a simple strategy based on Round-Robin selection of labeling requests. The statistical test performed on the results confirmed that our proposed active Twitter filtering method, ACTIF, outperforms the baselines, which shows that involving the user in the process significantly improves the performance of the filtering systems and that proposed active strategies are successful in selecting labeling requests in a way that increases the gain from user supervision.

4.5.1 Limitations and Future Work

There are several possible avenues for extending this work. We list them below.

1. Analyzing the impact of employing visualization techniques for our proposed tweet filtering approach. A research question is to what extent using visualization techniques enhances user supervision by reducing user effort or providing transparency of the filtering process, which in turn can improve the accuracy of the results and increase user satisfaction with the results.

2. Incorporating Twitter-specific features. We would like to study the effects of including specific features of Twitter, such as hashtags, on the effectiveness of selection strategies and the performance of filtering systems. Also, in ranking relevant tweets, we considered two features: their LM-based relevance score and the number of followees of their authors. Other features such as the appearance of URL(s) and/or hashtags, length of the tweets and the authors' locations can be considered for ranking tweets.

3. Evaluating the performance of pseudo-relevance feedback for automatic query expansion. The effectiveness of this technique is correlated with the accuracy of top recommended tweets. It has been demonstrated that the top two tweets that are returned by our retrieval model are not always relevant (see Table 4.5, profile "MB243"). In addition, some profiles have a few or no relevant tweets with regard to their actual relevance score. Therefore, it is expected that pseudo-relevance feedback would not perform satisfactorily, at least for the dataset used in this chapter, TRECMicroB dataset, with a low ratio of relevant tweets. However, further analysis is required.

4. Modifying the evaluation metric for silent days. We discussed that nDCG-1 is sensitive to the identification of silent days and nDCG-0 does not allocate any credit for the systems for silent days (see Section 4.4.2). As a future work, we intend to modify the nDCG metric for silent days in a way that penalizes each retrieval up to position k, where the system does not receive any credit after that. In this case, the system that remains silent in a silent day receives the perfect score, while systems with 1 to k retrieved tweets have nDCG values in

range (0,1), and after that the system does not receive any credit, nDCG =0. Investigating the effects of such a metric on the ranking of methods is another direction of future work.

5. Using external knowledge sources to find all alternative words that are used to refer to the same named entity. The experiments demonstrated that expanding named entities with their semantically related terms adds irrelevant terms to the query. However, sometimes other words are used to refer to the same named entity. For instance, using shortened names of places (e.g. cities, countries, airports) are very common in tweets. "CA" for "California", "UK" for "United Kingdom", and "JFK" for "John F. Kennedy" airport are some examples of using abbreviations for named entities. In addition, it is common to use nicknames for celebrities. Identifying these cases help filtering systems find more relevant tweets, which in turn improves their performance.

6. Evaluating the proposed strategies with other datasets. Since the proposed strategies are not limited to Twitter, we would like to apply them to other datasets (not tweets) to better understand the advantages or limitation of the proposed methods.

7. Assigning impact scores to labeling requests indicating their probability of improving the results. It was shown that active query expansion improves the average of the evaluation metrics over 51 topics in the test set. However, detailed results in Appendix J indicate that the query expansion is more successful in improving the results for some profiles than for others. An analysis of the results of individual profiles can help us prioritize profiles with respect to their estimated level of improvement by query expansion, which in turn enables us to manage user involvement in a more efficient way. In addition, query terms may also have different probabilities of being good candidates for query expansion. For instance, we did not consider named entities in the query expansion due to our hypothesis that expanding named entities may result in adding tweets that are not relevant to user interest. Proposing other strategies for estimating whether each query term should be further expanded is another interesting research problem.

8. Considering sentiment of tweets in the filtering process. Performing sentiment analysis and incorporating the sentiment of tweets in the filtering and recommnder systems might help in improving the quality of the recommended tweets. For instance, when the user is interested in official news about "legalizing medical marijuana", tweets that only contain personal opinions about this topic are not relevant to the user interest, while for a user who is interested in other passengers' opinions about "bus service to NYC" with regard to different aspects such as cleanliness, reliability and safety, opinionated tweets about this topic are interesting for the user.

9. Taking into account the relations between tweets in selection strategies. The proposed selection strategies select tweets incrementally, where the selection of particular tweets would affect the selection of other tweets as labeling requests. We considered two types of relations between tweets when selecting the labeling requests. One relation is that no near-duplicate tweets should exist among selected labeling requests. The other relation between labeling request is implicitly considered in Algorithm 5, which is when a candidate tweet contains a semantically related term, which has already been added to the excluded semantic group because of another labeling request containing the same term, the algorithm does not select that tweet as a labeling request. As a future work, we intend to study other types of relations between tweets, such as reply or retweet relations, when selecting labeling requests.

10. Selecting tweets from clusters based on their informativeness. In the novelty verification step, we selected the tweet which is published earliest within each cluster to be included in the final ranked list. Another strategy would be to select the tweet containing more information than other tweets in the same cluster, where the informativeness of tweets can be estimated based on various features such as the appearance of URL(s) and/or hashtags, length of their texts, and the authority of their authors.

11. Filtering untrustworthy tweets from retrieved relevant tweets. There are different solutions for the popular problems of rumor detection, identifying unreliable tweets, and spam detection. Applying the existing methods to identify such

tweets in order to filter them out from the results of recommendation and filtering systems, and analyzing their impact on user satisfaction is an interesting research direction.

Chapter 5

Conclusions

In recent years, more and more users have participated in generating and sharing online content rather than just consuming what is created by a limited number of publishers, which has resulted in a large volume of user-generated content. This type of data contains important information that can help businesses managers, journalists, sociologists, politicians and other professionals in making better decisions. Many forms of user-generated content are mainly text, such as customer reviews, blog comments, Facebook comments and tweets. Moreover, the scale, structure, length and semantics of this type of data is often different from traditional documents, which are often well-structured and written in a more formal fashion by following syntactic and grammatical rules. User-generated contents are usually short in size, contain spelling, syntactic and grammatical errors, and many acronyms and slang terms. Therefore, the performance of standard NLP techniques and traditional text mining methods deteriorates when applied to user-generated data.

One approach is to integrate visualization, active learning and user interaction techniques with text analytics methods to overcome the challenges of the analysis of large-scale user-generated data. These techniques have one similar goal: to involve the user in the analytical process in order to 1) benefit from her knowledge in improving the performance of the system, and 2) to augment the user cognitive and analytical skills in exploring the data, extracting relevant information, and making better decisions. Applications of user-generated content analysis cover a wide range of tasks from the topical classification of real-time data for improving crisis management to gender identification on social media for advertisement and personalized recommendation. In addition, due to the diversity of sources and types of user-generated data with different characteristics, gold standard datasets for benchmarking and comprehensive evaluations are not usually available. Consequently, the variety of the application and data types, the ever increasing number of online users

with different information needs, and the difficulties of measuring user satisfaction are reasons that the effectiveness of visualization and active learning techniques and the benefits of user involvement in the analysis of user-generated content has not yet been thoroughly studied. In this thesis, we investigated the effectiveness of user involvement in enhancing the performance of text mining techniques when applied to three challenging tasks on user-generated content by following our general framework in Chapter 1.

One popular type of user-generated content is customer reviews. We proposed a visual interface customized for the task of context-specific sentiment lexicon generation from product reviews. While sentiment lexicons play an important role in the accurate determination of sentiment in documents, creating such lexicons for every domain is an expensive task. The main objective of our proposed visual interface was to facilitate user supervision. We performed a user study, where the participants found the visual interface helpful in the task. We even found that the quality of the generated lexicon with visual interface is significantly better compared to a text-based interface, when participants are performing the task for the first time.

Twitter is another source of user-generated content containing important information that, if analyzed correctly, can help many users obtain better insight about their topics of interest. Our first problem was associating tweets with a set of topics. Based on Twitter-specific features, such as hashtags and the reply structure between tweets, we proposed active learning techniques for selecting labeling requests in a way that increases the recall and precision of the tweet retrieval. The proposed strategies outperformed the baselines, including a state-of-the-art active retrieval algorithm (ReQ-ReC), in improving the accuracy of the retrieval. In addition, an interactive visual interface, to assist users to better understand the retrieval process and selection strategies, was implemented. The evaluation of the interface with domain experts demonstrated the suitability of our proposed visual interface for making the process of associating tweets with different debates transparent, and the usefulness of the provided interactions for applying user feedback.

We considered the task of filtering tweets based on user interest profiles, where systems return a list of relevant tweets that are ranked based on their relevance to the user profile, as our final study. More particularly, we focused on expanding the

query using semantic relatedness methods in order to improve the performance of filtering systems by finding more relevant tweets. The experimental results confirmed that unsupervised expansion of the query does not significantly increase the quality of the recommended tweets. Consequently, we proposed active learning techniques that select instances to be labeled based on the objective of expanding the query with related terms. The comparison with a general supervised method, i.e. SVMRank, supported our hypothesis that if selection strategies are designed for a particular task, query expansion in this case, they better benefit from the user feedback.

In each chapter, we tried to answer two general research questions: 1) to what level the performance of text analytics methods is improved by user involvement in different tasks on user-generated data, and 2) whether integrating active learning or visualization techniques results in more efficient user interactions. This thesis answered these questions through three different tasks, which all reached the same conclusions: 1) the integration of user domain knowledge helps systems overcome the challenges of analyzing user-generated data and significantly improves the accuracy of the generated results by automatic methods, and 2) employing active learning and visualization techniques enables systems to increase their benefits from the user knowledge, enhances user supervision by reducing effort, and assists users to gain better insights.

5.1 Future Research Directions

In addition to the future work discussed in each chapter, which were specific to each problem, we hereby discuss more general research directions that can be pursued.

In all proposed methods for the aforementioned tasks, no stopping criterion was determined as we assumed that users would continue to interact with the system as long as they are not satisfied with the results or some limits are not reached (e.g. limit for the number of labeling requests). It might be useful to provide cues for users based on current status of the system. For instance, systems can provide users with an estimation of their performance, or highlight the changes in the results after each interaction through visualization techniques. When active learning techniques are employed, it is expected to observe major changes in the results after the first user interactions and smaller changes as the user continues. Providing this information to

the user might help her decide which interactions to focus on and when to stop. We implemented a small example of these techniques in our ATR-Vis visual interface by updating the number of retrieved tweets after each interaction and also highlighting newly retrieved tweets for each debate. Comprehensive analysis and evaluation of different stopping criteria is required. Another area of interest is to perform a user study on how the user decides to stop the supervision process in practice and how many questions she is willing to label.

Moreover, users, including non-technical persons and domain experts, make mistakes in their supervision. It is important to provide functions, which enable users to revert their actions in the interactive visualizations. This capability is provided in the proposed visual interfaces in Chapters 2 and 3. However, when the users are not aware of their mistakes, it would be beneficial if the system could provide cues based on the inconsistencies in the results or different feedback from the user. In addition, further analysis of the sensitivity of the performance of active learning techniques to users' mistakes is required. Furthermore, it is important to note the difference between user mistakes and user preferences. Sometimes, users provide different feedback because they have different preferences. For instance, a user may be interested in reading all tweets related to violence against women, while another user is only interested in tweets reporting these incidents in her local area, and therefore she may consider tweets that are labeled as relevant by the former user as irrelevant. Neither of these users are incorrect in labeling, but they have different interests. Therefore, another interesting yet challenging research direction might be to determine when users make mistakes.

One of the objectives of this thesis was to reduce user supervision effort in the analytical processing of data. Therefore, it is important to measure user effort carefully. We estimated user effort first by the number of labeling requests, and secondly through questionnaires, where users comment on whether they find the task easy or difficult. It would be interesting to propose a metric and a framework for quantifying the intensity of user involvement.

Finally, time is an important factor with the user-generated content. One of our motivations for the studies in this thesis was that as data and user needs change over time, models trained on old data cannot fulfill users' expectations anymore. It

would be useful to further evaluate proposed techniques on datasets with noticeable changes of some aspects of the data over time in order to examine whether the same conclusions can be made about these techniques or whether they should be modified according to the changes in the dataset.

Bibliography

[1] Ahmad Abdel-Hafez, Quoc Viet Phung, and Yue Xu. Utilizing voting systems for ranking user tweets. In *Proceedings of the 2014 Recommender Systems Challenge*, page 23, 2014.

[2] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Semantic enrichment of Twitter posts for user profile construction on the social web. In *Extended Semantic Web Conference*, pages 375–389. Springer, 2011.

[3] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Semantics+ filtering+ search= twitcident. exploring information in social web streams. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, pages 285–294. ACM, 2012.

[4] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of Twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics, 2011.

[5] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of NAACL HLT*, pages 19–27, 2009.

[6] Noa Aharony. Twitter use by three political leaders: an exploratory analysis. *Online Information Review*, 36(4):587–603, 2012.

[7] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. *International AAAI Conference on Web and Social Media*, 13:2–11, 2013.

[8] M Albakour, Craig Macdonald, Iadh Ounis, et al. On sparsity and drift for effective real-time filtering in microblogs. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 419–428. ACM, 2013.

[9] Aretha B Alencar, Maria Cristina F de Oliveira, and Fernando V Paulovich. Seeing beyond reading: a survey on visual text analytics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):476–492, 2012.

[10] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[11] Michelle Annett and Grzegorz Kondrak. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Proceedings of the 21st Conference on Advances in Artificial Intelligence*, pages 25–35, 2008.

[12] Dolan Antenucci, GREGORY Handy, AKSHAY Modi, and Miller Tinkerhess. Classification of tweets via clustering of hashtags. *EECS 545 Final Project*, 545:1–11, 2011.

[13] Richard Arias-Hernandez, Linda T Kaastra, Tera M Green, and Brian Fisher. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *44th Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2011.

[14] Muhammad Zubair Asghar, Shakeel Ahmad, Maria Qasim, Syeda Rabail Zahra, and Fazal Masud Kundi. Sentihealth: creating health-related sentiment lexicon using hybrid approach. *SpringerPlus*, 5(1):1139, 2016.

[15] Wouter Bancken, Daniele Alfarone, and Jesse Davis. Automatically detecting and rating product aspects from textual customer reviews. In *Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing at ECML/PKDD*, pages 1–16, 2014.

[16] Banjo. Banjo. `http://ban.jo/`. Accessed: March 2017.

[17] Andreas Bauer. *Information filtering in high velocity text streams using limited memory: an event-driven approach to text stream analysis*. PhD thesis, University of Regensburg, 2016.

[18] Nicholas J Belkin and W Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[19] Michael S Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam, and Ed H Chi. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, pages 303–312. ACM, 2010.

[20] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[21] Kalina Bontcheva, Genevieve Gorrell, and Bridgette Wessels. Social media and information overload: Survey results. *arXiv preprint arXiv:1306.0813*, 2013.

[22] Javier Borge-Holthoefer, Alejandro Rivero, Iñigo García, Elisa Cauhé, Alfredo Ferrer, Darío Ferrer, David Francos, David Iñiguez, María Pilar Pérez, Gonzalo Ruiz, et al. Structural and dynamical patterns on online social networks: the Spanish May 15th movement as a case study. *PloS one*, 6(8):e23883, 2011.

[23] Harald Bosch, Dennis Thom, Florian Heimerl, Edwin Puttmann, Steffen Koch, Robert Kruger, Michael Worner, and Thomas Ertl. Scatterblogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2022–2031, 2013.

[24] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. Tasteweights: a visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 35–42. ACM, 2012.

[25] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[26] Mike Bostock, Jeffrey Heer, Ogievetsky Vadim, and community. d3.js, data-driven documents. `http://d3js.org/`. Accessed: January 2013.

[27] S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. Learning document-level semantic properties from free-text annotations. *J. Artif. Int. Res.*, 34(1):569–603, April 2009.

[28] Thorsten Brants and Alex Franz. Web 1t 5-gram corpus version 1.1. *Google Inc*, 2006.

[29] Michael Brooks, John J Robinson, Megan K Torkildson, Cecilia R Aragon, et al. Collaborative visual analysis of sentiment in Twitter events. In *International Conference on Cooperative Design, Visualization and Engineering*, pages 1–8. Springer, 2014.

[30] Jürgen Broß and Heiko Ehrig. Generating a context-aware sentiment lexicon for aspect-based product review mining. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 435–439. IEEE, 2010.

[31] Pedro Henrique Calais Guerra, Adriano Veloso, Wagner Meira Jr, and Virgílio Almeida. From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–158. ACM, 2011.

[32] Rafael A Calvo and Sunghwan Mac Kim. Emotions in text: dimensional and categorical models. *Computational Intelligence*, 29(3):527–543, 2013.

[33] Amparo E Cano, Andrea Varga, Matthew Rowe, Fabio Ciravegna, and Yulan He. Harnessing linked knowledge sources for topic classification in social media. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 41–50. ACM, 2013.

[34] Nan Cao, Lu Lu, Yu-Ru Lin, Fei Wang, and Zhen Wen. Socialhelix: visual analysis of sentiment divergence in social media. *Journal of Visualization*, 18(2):221–235, 2015.

[35] Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. Interactive Multimedia Summaries of Evaluative Text. In *Proceedings of the 11th International Conference on IUI*, pages 124–131, New York, NY, USA, 2006. ACM.

[36] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1, 2012.

[37] Andrea Ceron, Luigi Curini, Stefano M Iacus, and Giuseppe Porro. Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens political preferences with an application to italy and france. *New Media & Society*, 16(2):340–358, 2014.

[38] Junghoon Chae, Dennis Thom, Harald Bosch, Yun Jang, Ross Maciejewski, David S Ebert, and Thomas Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *IEEE Conference on Visual Analytics Science and Technology*, pages 143–152. IEEE, 2012.

[39] Chaomei Chen, Fidelia Ibekwe-SanJuan, Eric SanJuan, and Chris Weaver. Visual analysis of conflicting opinions. In *IEEE Symposium On Visual Analytics Science and Technology*, pages 59–66. IEEE, 2006.

[40] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 201–210. ACM, 2009.

[41] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1185–1194. ACM, 2010.

[42] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 661–670, 2012.

[43] Yan Chen, Zhoujun Li, Liqiang Nie, Xia Hu, Xiangyu Wang, Tat-seng Chua, and Xiaoming Zhang. A semi-supervised bayesian network model for microblog topic classification. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 561–576. Citeseer, 2012.

[44] Yu-Sheng Chen, Lieu-Hen Chen, and Yasufumi Takama. Proposal of lda-based sentiment visualization of hotel reviews. In *IEEE International Conference on Data Mining Workshop*, pages 687–693. IEEE, 2015.

[45] Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on EMNLP: Volume 2*, pages 590–598, Stroudsburg, PA, USA, 2009. ACL.

[46] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 443–452. ACM, 2012.

[47] Peter Cogan, Matthew Andrews, Milan Bradonjic, W Sean Kennedy, Alessandra Sala, and Gabriel Tucci. Reconstruction and analysis of Twitter conversation graphs. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, pages 25–31. ACM, 2012.

[48] Michael Conover, Jacob Ratkiewicz, Matthew R Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on Twitter. *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, 133:89–96, 2011.

[49] Danish Contractor, Bhupesh Chawda, Sameep Mehta, L Venkata Subramaniam, and Tanveer A Faruquie. Tracking political elections on social media: applications and experience. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2320–2326. AAAI Press, 2015.

[50] Gordon V Cormack and Maura R Grossman. Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1039–1048. ACM, 2016.

[51] Bruce Croft and John Lafferty. *Language modeling for information retrieval*, volume 13. Springer Science & Business Media, 2013.

[52] Anh Dang, Raheleh Makki, Abidalrahman Moh'd, Aminul Islam, Vlado Keselj, and Evangelos E Milios. Real time filtering of tweets using Wikipedia concepts and Google tri-gram semantic relatedness. Technical report, Dalhousie University Halifax, NS Canada, 2015.

[53] DBpedia. DBpedia. `http://wiki.dbpedia.org/`. Accessed: March 2017.

[54] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *In Proceedings of International Conference on Language Resources and Evaluation*, volume 6, pages 449–454. Genoa, 2006.

[55] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

[56] Shuyuan Deng, Atish P Sinha, and Huimin Zhao. Adapting sentiment lexicons to domain-specific social media texts. *Decision Support Systems*, 2016.

[57] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Recent Advances in Natural Language Processing*, pages 198–206, 2013.

[58] Mitali Desai and Mayuri A Mehta. Techniques for sentiment analysis of Twitter data: A comprehensive survey. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*, pages 149–154. IEEE, 2016.

[59] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 115–122. IEEE, 2010.

[60] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. Association for Computational Linguistics, 2012.

[61] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM, 2008.

[62] Marian Dörk, Daniel Gruen, Carey Williamson, and Sheelagh Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

[63] Wenwen Dou, Xiaoyu Wang, Drew Skau, William Ribarsky, and Michelle X Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *IEEE Conference on Visual Analytics Science and Technology*, pages 93–102. IEEE, 2012.

[64] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

[65] Javid Ebrahimi, Dejing Dou, and Daniel Lowd. Weakly supervised tweet stance classification by relational bootstrapping. In *Proceedings of EMNLP*, 2016.

[66] Miles Efron. Hashtag retrieval in a microblogging environment. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788. ACM, 2010.

[67] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation*, pages 417–422, 2006.

[68] Facebook. Facebook. `https://www.facebook.com/`. Accessed: March 2017.

[69] Angela Fahrni and Manfred Klenner. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proceedings of the Symposium on Affective Language in Human and Machine*, pages 60–63, 2008.

[70] Xing Fang and Justin Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5, 2015.

[71] Wei Feng and Jianyong Wang. We can learn your# hashtags: Connecting tweets to explicit topics. In *30th IEEE International Conference on Data Engineering*, pages 856–867. IEEE, 2014.

[72] Yue Feng, Hossein Fani, Ebrahim Bagheri, and Jelena Jovanovic. Lexical semantic relatedness for Twitter analytics. In *27th International Conference on Tools with Artificial Intelligence*, pages 202–209. IEEE, 2015.

[73] Andy Field. *Discovering statistics using SPSS*. Sage publications, 2009.

[74] Renato Miranda Filho, Jussara M. Almeida, and Gisele L. Pappa. Twitter population sample bias and its impact on predictive outcomes: A case study on elections. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1254–1261. ACM, 2015.

[75] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd ACL*, pages 363–370, 2005.

[76] The Apache Software Foundation. Lucene. `https://lucene.apache.org/`. Accessed: March 2017.

[77] Isvani Frías-Blanco, Alberto Verdecia-Cabrera, Agustín Ortiz-Díaz, and Andre Carvalho. Fast adaptive stacking of ensembles. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 929–934. ACM, 2016.

[78] Lorenzo Gabrielli, Salvatore Rinzivillo, Francesco Ronzano, and Daniel Villatoro. From tweets to semantic trajectories: mining anomalous urban mobility patterns. In *Citizen in Sensor Networks*, pages 26–35. Springer, 2014.

[79] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

[80] Devin Gaffney. #iranelection: Quantifying online activism. In *Proceedings of WebSci10: Extending the Frontiers of Society On-Line*, 2010.

[81] Philippe Gambette and Jean Véronis. Visualising a text with a tree cloud. pages 561–569, 2010.

[82] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. Pulse: Mining Customer Opinions from Free Text. In *Proceedings of the 6th International Conference on Advances in Intelligent Data Analysis*, IDA'05, pages 121–132, 2005.

[83] Murthy Ganapathibhotla and Bing Liu. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 241–248. Association for Computational Linguistics, 2008.

[84] Dehong Gao, Furu Wei, Wenjie Li, Xiaohua Liu, and Ming Zhou. Cross-lingual sentiment lexicon learning with bilingual word graph label propagation. *Computational Linguistics*, 2015.

[85] Sandra Garcia Esparza, Michael P O'Mahony, and Barry Smyth. Catstream: categorising tweets for user profiling and stream filtering. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 25–36. ACM, 2013.

[86] Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. Entity extraction, linking, classification, and tagging for social media: a Wikipedia-based approach. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013.

[87] Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. Entity extraction, linking, classification, and tagging for social media: a Wikipedia-based approach. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013.

[88] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V Nickerson. Discovering context: classifying tweets through a semantic transform based on Wikipedia. In *International Conference on Foundations of Augmented Cognition*, pages 484–492. Springer, 2011.

[89] M Rami Ghorab, Dong Zhou, Alexander OConnor, and Vincent Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, 2013.

[90] Anastasia Giachanou and Fabio Crestani. Like it or not: A survey of Twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):28, 2016.

[91] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[92] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for Twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 593–596. ACM, 2013.

[93] Mona Golestan Far, Scott Sanne, Mohamed Reda Bouadjenek, Gabriela Ferraro, and David Hawking. On term selection techniques for patent prior art search. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 803–806. ACM, 2015.

[94] Travis Goodwin and Sanda M Harabagiu. UTD at TREC 2014: Query expansion for clinical decision support. Technical report, DTIC Document, 2014.

[95] Google. Freebase. `https://developers.google.com/freebase/`, 2017. Accessed: March 2017.

[96] Michelle L Gregory, Nancy Chinchor, Paul Whitney, Richard Carter, Elizabeth Hetzler, and Alan Turner. User-directed sentiment analysis: Visualizing the affective content of documents. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 23–30. Association for Computational Linguistics, 2006.

[97] Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Christopher Hall, and Tobias Höllerer. Smallworlds: visualizing social recommendations. In *Computer Graphics Forum*, volume 29, pages 833–842. Wiley Online Library, 2010.

[98] The Stanford Natural Language Processing Group. The stanford parser: A statistical parser. `http://nlp.stanford.edu/software/lex-parser.shtml`. Accessed: March 2017.

[99] Anatoliy Gruzd and Jeffrey Roy. Investigating political polarization on Twitter: A Canadian perspective. *Policy & Internet*, 6(1):28–45, 2014.

[100] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on Twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

[101] Davide F Gurini and Fabio Gasparetti. Trec microblog 2012 track: Real-time algorithm for microblog ranking systems. Technical report, DTIC Document, 2012.

[102] Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc top-n similarity. `http://swoogle.umbc.edu/SimService/top_similarity.html`. Accessed: April 2016.

[103] Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 44–52, 2013.

[104] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, 2001.

[105] Ming Hao, Daniel A Keim, Umeshwar Dayal, Daniela Oelke, and Chantal Tremblay. Density displays for data stream monitoring. In *Computer Graphics Forum*, volume 27, pages 895–902. Wiley Online Library, 2008.

[106] Ming C. Hao, Christian Rohrdantz, Halldor Janetzko, Daniel A. Keim, Umeshwar Dayal, Lars-Erik Haug, and Meichun Hsu. Integrating Sentiment Analysis and Term Associations with Geo-Temporal Visualizations on Customer Feedback Streams. In *SPIE 2012 Conference on Visualization and Data Analysis*, volume 8294, 2012.

[107] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *IEEE Symposium on Information Visualization*, pages 115–123. IEEE, 2000.

[108] HGI. Harvard General Inquirer. `http://www.wjh.harvard.edu/~inquirer`. Accessed: December 2012.

[109] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[110] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[111] Hootsuite. Hootsuite. `https://hootsuite.com/`. Accessed: March 2017.

[112] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177. ACM, 2004.

[113] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In *Association of the Advancement of Artificial Intelligence*, volume 4, pages 755–760, 2004.

[114] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 919–928. ACM, 2009.

[115] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Actnet: Active learning for networked texts in microblogging. In *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM13)*, pages 306–314, 2013.

[116] Yuheng Hu, Ajita John, Dorée Duncan Seligmann, and Fei Wang. What were the tweets about? topical associations between public events and Twitter feeds. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[117] Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. Collective tweet wikification based on semi-supervised graph regularization. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 380–390, 2014.

[118] Jeff Huang, Katherine M Thornton, and Efthimis N Efthimiadis. Conversational tagging in Twitter. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 173–178. ACM, 2010.

[119] Sheng-Jun Huang, Songcan Chen, and Zhi-Hua Zhou. Multi-label active learning: Query type matters. In *IJCAI*, pages 946–952, 2015.

[120] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *Advances in Neural Information Processing Systems*, pages 892–900, 2010.

[121] Zhao Huang and Morad Benyoucef. From e-commerce to social commerce: A close look at design features. *Electronic Commerce Research and Applications*, 12(4):246–259, 2013.

[122] Schuyler Huck. *Reading Statistics and Research*. Alyn & Bacon, Inc., 2008.

[123] Ajaz Hussain, Khalid Latif, Aimal Tariq Rextin, Amir Hayat, and Masoon Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355–367, 2014.

[124] Elena Ilina, Claudia Hauff, Ilknur Celik, Fabian Abel, and Geert-Jan Houben. Social event detection on Twitter. In *Web Engineering*, pages 169–176. Springer, 2012.

[125] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. Aidr: Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 159–162. ACM, 2014.

[126] Aminul Islam, Evangelos Milios, and Vlado Kešelj. Text similarity using Google tri-grams. In *Canadian Conference on Artificial Intelligence*, pages 312–317. Springer, 2012.

[127] Tommi Jaakkola and Hava Siegelmann. Active information retrieval. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 777–784. 2001.

[128] Harrie Jansen. The logic of qualitative survey research and its position in the field of social research methods. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, volume 11, 2010.

[129] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[130] Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594. Association for Computational Linguistics, 2010.

[131] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226. ACM, 2006.

[132] Daniel Jurafsky and H James. *Speech and language processing an introduction to natural language processing, computational linguistics, and speech*. Pearson Education,, 2000.

[133] Jari Jussila, Jukka Huhtamäki, Hannu Kärkkäinen, and Kaisa Still. Information visualization of Twitter data for co-organizing conferences. In *Proceedings of International Conference on Making Sense of Converging Media*, pages 139–145. ACM, 2013.

[134] Jan Kalmeijer. Hashtag clustering to summarize the topics discussed by dutch members of parliament. volume 8, page 10, 2014.

[135] Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of 4th International Conference on Language Resources and Evaluation*, pages 1115–1118, 2004.

[136] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics, 2006.

[137] Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, and Amit Sheth. User interests identification on Twitter using a hierarchical knowledge base. In *European Semantic Web Conference*, pages 99–113. Springer, 2014.

[138] Noriaki Kawamae. Supervised n-gram topic model. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 473–482. ACM, 2014.

[139] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the information age solving problems with visual analytics*. Eurographics Association, 2010.

[140] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual analytics: Scope and challenges. In *Visual Data Mining*, pages 76–90. Springer, 2008.

[141] Renato Kempter, Valentina Sintsova, Claudiu Cristian Musat, and Pearl Pu. Emotionwatch: Visualizing fine-grained emotions in event-related tweets. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[142] Tom Kenter and Maarten de Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM, 2015.

[143] Wiltrud Kessler, Roman Klinger, and Jonas Kuhn. Towards opinion mining from reviews for the prediction of product rankings. In *6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, page 51, 2015.

[144] Soo-Min Kim and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pages 483–490, Stroudsburg, PA, USA, 2006. ACL.

[145] Andy Kirk. *Data Visualization: a successful design process*. Packt Publishing Ltd, 2012.

[146] Ahmet Koçyiğit, Dilek Tapucu, Berrin Yanikoglu, Yücel Saygın, et al. An aspect-lexicon creation and evaluation tool for sentiment analysis researchers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 804–807. Springer, 2012.

[147] Kostiantyn Kucher, Teri Schamp-Bjerede, Andreas Kerren, Carita Paradis, and Magnus Sahlgren. Visual analysis of online social media to open up the investigation of stance phenomena. *Information Visualization*, 15(2):93–116, 2016.

[148] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more?: the effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2012.

[149] Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi, and Huan Liu. Tweettracker: An analysis tool for humanitarian and disaster relief. In *International AAAI Conference on Weblogs and Social Media*, pages 661–662, 2011.

[150] Su Mon Kywe, Ee-Peng Lim, and Feida Zhu. A survey of recommender systems in Twitter. In *International Conference on Social Informatics*, pages 420–433. Springer, 2012.

[151] Parisa Lak, Mefta Sadat, Carl Julien Barrelet, Martin Petitclerc, Andriy Miranskyy, Craig Statchuk, and Ayse Basar Bener. Preliminary investigation on user interaction with ibm watson analytics. 2016.

[152] Heidi Lam, Enrico Bertini, Petra Isenberg, Catherine Plaisant, and Sheelagh Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, 2012.

[153] Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. Twitter trending topic classification. In *11th IEEE International Conference on Data Mining Workshops*, pages 251–258. IEEE, 2011.

[154] Dados Abertos Legislativos. Dados abertos legislativos. `http://dadosabertos.senado.gov.br/`. Accessed: May 2015.

[155] Cheng Li, Yue Wang, Paul Resnick, and Qiaozhu Mei. Req-rec: High recall retrieval with query pooling and interactive classification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 163–172. ACM, 2014.

[156] Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th Annual Meeting of the ACL: Long Papers - Volume 1*, pages 410–419, Stroudsburg, PA, USA, 2012. ACL.

[157] Jimmy Lin. Twitter Tools. `https://github.com/lintool/twitter-tools`. Accessed: July 2015.

[158] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, and Ellen Voorhees. Overview of the TREC-2015 microblog track. In *Proceedings of the TREC*, 2015.

[159] LinkedIn. LinkedIn. `https://www.linkedin.com/`. Accessed: March 2017.

[160] B. Liu. Opinion mining, sentiment analysis, and opinion spam detection. `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`. Accessed: December 2012.

[161] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM, 2005.

[162] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.

[163] Mengchen Liu, Shixia Liu, Xizhou Zhu, Qinying Liao, Furu Wei, and Shimei Pan. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):250–259, 2016.

[164] Shixia Liu, Xiting Wang, Jianfei Chen, Jim Zhu, and Baining Guo. Topicpanorama: A full picture of relevant topics. In *IEEE Conference on Visual Analytics Science and Technology*, pages 183–192. IEEE, 2014.

[165] Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. Entity linking for tweets. In *ACL (1)*, pages 1304–1311, 2013.

[166] Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, and Xiangyang Zhou. Joint inference of named entity recognition and normalization for tweets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 526–535. Association for Computational Linguistics, 2012.

[167] Livefyre. Storify. `https://storify.com/`. Accessed: March 2017.

[168] Benedikt Loepp, Katja Herrmanny, and Jürgen Ziegler. Blended recommending: Integrating interactive information filtering and algorithmic recommender techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 975–984. ACM, 2015.

[169] Yafeng Lu, Robert Krüger, Dennis Thom, Feng Wang, Steffen Koch, Thomas Ertl, and Ross Maciejewski. Integrating predictive analytics and social media. In *IEEE Conference on Visual Analytics Science and Technology*, pages 193–202. IEEE, 2014.

[170] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, pages 347–356. ACM, 2011.

[171] William Lucia and Elena Ferrari. Egocentric: Ego networks for knowledge-based short text classification. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1079–1088. ACM, 2014.

[172] Zhunchen Luo, Miles Osborne, and Ting Wang. Opinion retrieval in Twitter. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, pages 507–510, 2012.

[173] Zhunchen Luo, Miles Osbornez, Saša Petrovic, and Ting Wang. Improving Twitter retrieval by exploiting structural information. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 648–654, 2012.

[174] Alan M MacEachren, Anuj Jaiswal, Anthony C Robinson, Scott Pezanowski, Alexander Savelyev, Prasenjit Mitra, Xiao Zhang, and Justine Blanford. Senseplace2: GeoTwitter analytics support for situational awareness. In *IEEE Conference on Visual Analytics Science and Technology*, pages 181–190. IEEE, 2011.

[175] Raheleh Makki. Active tweet recommendation based on user interest profiles. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media (Extended Proceedings)*, 2016.

[176] Raheleh Makki, Stephen Brooks, and Evangelos E Milios. Context-specific sentiment lexicon expansion via minimal user interaction. In *International Conference on Information Visualization Theory and Applications*, pages 178–186. IEEE, 2014.

[177] Raheleh Makki, Eder Carvalho, Axel J Soto, Stephen Brooks, Maria Cristina Ferreira De Oliveira, Evangelos E Milios, and Rosane Minghim. ATR-Vis: Visual and interactive information retrieval for parliamentary discussions in Twitter. *To appear in ACM Transactions on Knowledge Discovery from Data (TKDD), Accepted January 2017*, 2017.

[178] Raheleh Makki, Axel J Soto, Stephen Brooks, and Evangelos E Milios. Active information retrieval for linking Twitter posts with political debates. In *14th International Conference on Machine Learning and Applications (ICMLA)*, pages 238–245. IEEE, 2015.

[179] Raheleh Makki, Axel J Soto, Stephen Brooks, and Evangelos E Milios. Twitter message recommendation based on user interest profiles. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 406–410. IEEE, 2016.

[180] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[181] Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–236. ACM, 2011.

[182] Carlos Martin, David Corney, and Ayse Göker. Finding newsworthy topics on Twitter. *IEEE Computer Society Special Technical Community on Social Networking E-Letter*, 1(3), 2013.

[183] Eugenio Martínez-Cámara, M Teresa Martín-Valdivia, L Alfonso Urena-López, and A Rturo Montejo-Ráez. Sentiment analysis in Twitter. *Natural Language Engineering*, 20(01):1–28, 2014.

[184] Kamran Massoudi, Manos Tsagkias, Maarten De Rijke, and Wouter Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *European Conference on Information Retrieval*, pages 362–367. Springer, 2011.

[185] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pages 563–572. ACM, 2012.

[186] Qingliang Miao, Qiudan Li, and Ruwei Dai. Amazing: A sentiment mining and retrieval system. *Expert Systems with Applications*, 36(3):7192–7198, 2009.

[187] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 233–242. ACM, 2007.

[188] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[189] Dmitrijs Milajevs and Gosse Bouma. Real time discussion retrieval from Twitter. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pages 795–800. International World Wide Web Conferences Steering Committee, 2013.

[190] Violeta Mirchevska, Mitja Luštrek, and Matjaž Gams. Combining domain knowledge and machine learning for robust fall detection. *Expert Systems*, 31(2):163–175, 2014.

[191] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 439–448. ACM, 2013.

[192] Samaneh Moghaddam and Martin Ester. On the design of lda models for aspect-based opinion mining. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 803–812. ACM, 2012.

[193] Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. Semeval-2016 task 6: Detecting stance in tweets. *Proceedings of SemEval*, 16, 2016.

[194] Saif M Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *arXiv preprint arXiv:1605.01655*, 2016.

[195] Fred Morstatter, Shamanth Kumar, Huan Liu, and Ross Maciejewski. Understanding Twitter data with tweetxplorer. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1482–1485. ACM, 2013.

[196] MPQA. Multi-Perspective Question Answering. `http://www.cs.pitt.edu/mpqa/`. Accessed: December 2012.

[197] Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, 2014.

[198] Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 339–348. Association for Computational Linguistics, 2012.

[199] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[200] Preslav Nakov, Sara Rosenthal, Svetlana Kiritchenko, Saif M Mohammad, Zornitsa Kozareva, Alan Ritter, Veselin Stoyanov, and Xiaodan Zhu. Developing a successful semeval task in sentiment analysis of Twitter and other social media texts. *Language Resources and Evaluation*, 50(1):35–65, 2016.

[201] Chris North and Ben Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 128–135. ACM, 2000.

[202] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2, 2010.

[203] Brendan O'Connor, Michel Krieger, and David Ahn. Tweetmotif: Exploratory search and topic summarization for Twitter. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 384–385, 2010.

[204] John O'Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. Peerchooser: visual interactive recommendation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1085–1088. ACM, 2008.

[205] Parliament of Canada. Parliament of canada. `http://www.parl.gc.ca/Default.aspx?Language=E`. Accessed: May 2014.

[206] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. (1100-3154), 2009.

[207] Miles Osborne, Ashwin Lall, and Benjamin Van Durme. Exponential reservoir sampling for streaming language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 687–692, 2014.

[208] Mark Otto, Jacob Thornton, and Bootstrap contributors. Bootstrap, the world's most popular mobile-first and responsive front-end framework. `http://getbootstrap.com/`. Accessed: March 2017.

[209] Jiaul H Paik and Jimmy Lin. Do multiple listeners to the public Twitter sample stream receive the same tweets? In *SIGIR Workshop on Temporal, Social and Spatially-Aware Information Access*, 2015.

[210] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 45–54. ACM, 2011.

[211] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[212] Sungrae Park, Wonsung Lee, and Il-Chul Moon. Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recognition Letters*, 56:38–44, 2015.

[213] Victor Pascual-Cid and Andreas Kaltenbrunner. Exploring asynchronous online discussions through hierarchical visualisation. In *2009 13th International Conference Information Visualisation*, pages 191–196. IEEE, 2009.

[214] Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299, 2007.

[215] M-H Peetz, Damiano Spina, Julio Gonzalo, M Rijke, et al. Towards an active learning system for company name disambiguation in microblog streams. In *CEUR Workshop Proceedings*, number 1179. CEUR, 2013.

[216] Marco Pennacchiotti, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. Making your interests follow you on Twitter. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 165–174. ACM, 2012.

[217] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[218] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, pages 91–100. ACM, 2008.

[219] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. Terms of a feather: Content-based news recommendation and discovery using Twitter. In *European Conference on Information Retrieval*, pages 448–459. Springer, 2011.

[220] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using Twitter to recommend real-time topical news. In *Proceedings of the third ACM Conference on Recommender Systems*, pages 385–388. ACM, 2009.

[221] PoliTwitter. PoliTwitter. `http://politwitter.ca/page/canadian-politics-twitters/mp/house`. Accessed: May 2014.

[222] Philips Kokoh Prasetyo, Palakorn Achananuparp, and Ee-Peng Lim. On analyzing geotagged tweets for location-based patterns. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pages 45–50. ACM, 2016.

[223] Princeton University. Wordnet: A Lexical Database for English. `https://wordnet.princeton.edu/`, 2010. Accessed: March 2017.

[224] Runwei Qiang, Feifan Fan, Chao Lv, and Jianwu Yang. Knowledge-based query expansion in real-time microblog search. *arXiv preprint arXiv:1503.03961*, 2015.

[225] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In *IJCAI*, volume 9, pages 1199–1204, 2009.

[226] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27, 2011.

[227] Zhonghua Qu and Yang Liu. Interactive group suggesting for Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 519–523. Association for Computational Linguistics, 2011.

[228] Vineeth Rakesh, Dilpreet Singh, Bhanukiran Vinzamuri, and Chandan K Reddy. Personalized recommendation of Twitter lists using content and network information. In *International AAAI Conference on Web and Social Media*, 2014.

[229] Toqir A Rana and Yu-N Cheah. Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*, 46(4):459–483, 2016.

[230] Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–682. Association for Computational Linguistics, 2009.

[231] Kumar Ravi and Vadlamani Ravi. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46, 2015.

[232] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[233] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[234] Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

[235] Manuel Gomez Rodriguez, Krishna Gummadi, and Bernhard Schoelkopf. Quantifying information overload in social media and its impact on social contagions. *arXiv preprint arXiv:1403.6838*, 2014.

[236] Christian Rohrdantz, Ming C Hao, Umeshwar Dayal, Lars-Erik Haug, and Daniel A Keim. Feature-based visual sentiment analysis of text document streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):1–26, 2012.

[237] Daniel M Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, pages 695–704. ACM, 2011.

[238] Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM*, 2011.

[239] André Luis Debiaso Rossi, André Carlos Ponce de Leon Ferreira, Carlos Soares, Bruno Feres De Souza, et al. Metastream: A meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, 127:52–64, 2014.

[240] Mickael Rouvier and Benoit Favre. Building a robust sentiment lexicon with (almost) no resource. *arXiv preprint arXiv:1612.05202*, 2016.

[241] Eduardo J Ruiz, Vagelis Hristidis, and Panagiotis G Ipeirotis. Efficient filtering on hidden document streams. In *International AAAI Conference on Weblogs and Social Media*, 2014.

[242] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, 52(1):5–19, 2016.

[243] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

[244] Armin Sajadi, Evangelos E Milios, Vlado Kešelj, and Jeannette CM Janssen. Wikisim. `http://ares.research.cs.dal.ca/~sajadi/wikisim/`. Accessed: March 2017.

[245] Armin Sajadi, Evangelos E Milios, Vlado Kešelj, and Jeannette CM Janssen. Domain-specific semantic relatedness from Wikipedia structure: A case study in biomedical text. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 347–360. Springer, 2015.

[246] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860. ACM, 2010.

[247] Justin Sampson, Fred Morstatter, Ross Maciejewski, and Huan Liu. Surpassing the limit: Keyword clustering to improve Twitter sample coverage. In *Proceedings of the 26th ACM Conference on Hypertext and Social Media*, pages 237–245. ACM, 2015.

[248] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *Proceedings of the 17th ACM Sigspatial International Conference on Advances in Geographic Information Systems*, pages 42–51. ACM, 2009.

[249] Jesus Serrano-Guerrero, Jose A Olivas, Francisco P Romero, and Enrique Herrera-Viedma. Sentiment analysis: a review and comparative analysis of web services. *Information Sciences*, 311:18–38, 2015.

[250] David Shamma, Lyndon Kennedy, and Elizabeth Churchill. Tweetgeist: Can the Twitter timeline reveal the structure of broadcast events? *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, pages 589–593, 2010.

[251] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.

[252] Keyi Shen, Jianmin Wu, Ya Zhang, Yiping Han, Xiaokang Yang, Li Song, and Xiao Gu. Reorder user's tweets. *ACM Transactions on Intelligent Systems and Technology*, 4(1):6, 2013.

[253] Abhishek Sikchi, Pawan Goyal, and Samik Datta. Peq: An explainable, specification-based, aspect-oriented product comparator for e-commerce. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2029–2032. ACM, 2016.

[254] Malcolm Slaney and Michael Casey. Locality-sensitive hashing for finding nearest neighbors [lecture notes]. *IEEE Signal Processing Magazine*, 25(2):128–131, 2008.

[255] Axel J Soto, Ryan Kiros, Vlado Keselj, and Evangelos Milios. Machine learning meets visualization for extracting insights from text data. *AI Matters*, 2(2):15–17, 2016.

[256] Axel J Soto, Abidalrahman Mohammad, Andrew Albert, Aminul Islam, Evangelos Milios, Michael Doyle, Rosane Minghim, and Maria Cristina Ferreira de Oliveira. Similarity-based support for text reuse in technical writing. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 97–106. ACM, 2015.

[257] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in Twitter to improve information filtering. In *Proceedings of the 33rd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–842, 2010.

[258] Guodao Sun, Yingcai Wu, Shixia Liu, Tai-Quan Peng, Jonathan J. H. Zhu, and Ronghua Liang. Evoriver: Visual analysis of topic competition on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1753–1762, 2014.

[259] Shiliang Sun, Chen Luo, and Junyu Chen. A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36:10–25, 2017.

[260] Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting Semantic Orientations of Words Using Spin Model. In *Proceedings of the 43rd Annual Meeting on ACL*, pages 133–140, Stroudsburg, PA, USA, 2005.

[261] Luchen Tan, Adam Roegiest, Charles LA Clarke, and Jimmy Lin. Simple dynamic emission strategies for microblog filtering. In *Proceedings of the 39th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1009–1012, 2016.

[262] Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles LA Clarke. An exploration of evaluation metrics for mobile push notifications. In *Proceedings of the 39th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 741–744, 2016.

[263] Jaime Teevan, Susan T Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 449–456. ACM, 2005.

[264] James J Thomas, Kristin Cook, et al. A visual analytics agenda. *IEEE Computer Graphics and Applications*, 26(1):10–13, 2006.

[265] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *23rd IEEE International Conference on Data Engineering Workshop*, pages 801–810. IEEE, 2007.

[266] Text Retrieval Conference (TREC). Trec 2015 microblog track judgement file. `http://trec.nist.gov/data/microblog/2015/qrels.txt`. Accessed: APril 2016.

[267] Text Retrieval Conference (TREC). Trec 2015 microblog track test topics. `http://trec.nist.gov/data/microblog/2015/TREC2015-MB-testtopics.txt`. Accessed: APril 2016.

[268] Text Retrieval Conference (TREC). Trec real-time summarization track homepage. `trecrts.github.io/TREC2016-RTS-guidelines.html`. Accessed: March 2017.

[269] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment. *International AAAI Conference on Weblogs and Social Media*, 10(1):178–185, 2010.

[270] Peter D Turney and Michael L Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

[271] Twitter. Twitter. `https://about.twitter.com/company`. Accessed: March 2017.

[272] Twitter. Twitter Streaming APIs. `https://dev.twitter.com/streaming/overview`. Accessed: May 2014.

[273] Ibrahim Uysal and W Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 2261–2264. ACM, 2011.

[274] Cuong Van Tran, Tuong Tri Nguyen, Dinh Tuyen Hoang, Dosam Hwang, and Ngoc Thanh Nguyen. Active learning-based approach for named entity recognition on short text streams. In *Multimedia and Network Information Systems*, pages 321–330. Springer, 2017.

[275] Kasturi Dewi Varathan, Anastasia Giachanou, and Fabio Crestani. Comparative opinion mining: a review. *Journal of the Association for Information Science and Technology*, 2016.

[276] Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links. In *International Conference on Database Systems for Advanced Applications*, pages 397–413. Springer, 2012.

[277] Wesley Waldner and Julita Vassileva. Emphasize, don't filter!: displaying recommendations in Twitter timelines. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 313–316. ACM, 2014.

[278] Wesley Waldner and Julita Vassileva. A visualization interface for Twitter timeline activity. In *Joint Workshop on Interfaces and Human Decision Making in Recommender Systems*, page 45, 2014.

[279] Changbo Wang, Zhao Xiao, Yuhua Liu, Yanru Xu, Aoying Zhou, and Kang Zhang. Sentiview: Sentiment analysis and visualization for internet popular topics. *IEEE Transactions on Human-Machine Systems*, 43(6):620–630, 2013.

[280] Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. Locally weighted linear regression for cross-lingual valence-arousal prediction of affective words. *Neurocomputing*, 194:271–278, 2016.

[281] Yu Wang, Eugene Agichtein, and Michele Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 123–131. ACM, 2012.

[282] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119. ACM, 2000.

[283] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012.

[284] Andrew Webster and Julita Vassileva. The keepup recommender system. In *Proceedings of the ACM conference on Recommender systems*, pages 173–176. ACM, 2007.

[285] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 153–162. ACM, 2010.

[286] Albert Weichselbraun, Stefan Gindl, and Arno Scharl. A context-dependent supervised learning approach to sentiment detection in large textual databases. *Journal of Information and Data Management*, 1(3):329–342, 2010.

[287] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[288] Wikipedia. Wikipedia, the free encyclopedia. `https://en.wikipedia.org/`, 2011. Accessed: March 2017.

[289] word2vec. word2vec. `https://code.google.com/archive/p/word2vec/`. Accessed: March 2016.

[290] Yingcai Wu, Shixia Liu, Kai Yan, Mengchen Liu, and Fangzhao Wu. Opinionflow: Visual analysis of opinion diffusion on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1763–1772, 2014.

[291] Yingcai Wu, Furu Wei, Shixia Liu, Norman Au, Weiwei Cui, Hong Zhou, and Huamin Qu. Opinionseer: Interactive Visualization of Hotel Customer Feedback. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1109–1118, November 2010.

[292] Jinxi Xu and W Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11. ACM, 1996.

[293] Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. Mining comparative opinions from customer reviews for competitive intelligence. *Decision Support Systems*, 50(4):743–754, 2011.

[294] Panpan Xu, Yingcai Wu, Enxun Wei, Tai-Quan Peng, Shixia Liu, J.J.H. Zhu, and Huamin Qu. Visual analysis of topic competition on social media. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2012–2021, Dec 2013.

[295] Wei Xu, Chris Callison-Burch, and William B Dolan. Semeval-2015 task 1: Paraphrase and semantic similarity in Twitter (pit). *Proceedings of the 9th International Workshop on Semantic Evaluation*, page 111, 2015.

[296] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. Writer meets reader: Emotion analysis of social media from both the writer's and reader's perspectives. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, volume 1, pages 287–290. IEEE, 2009.

[297] Shuang-Hong Yang, Alek Kolcz, Andy Schlaikjer, and Pankaj Gupta. Large-scale high-precision topic modeling on Twitter. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1907–1916. ACM, 2014.

[298] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE International Conference on Data Mining*, pages 427–434. IEEE, 2003.

[299] Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K Robert Lai, and Xuejie Zhang. Building chinese affective resources in valence-arousal dimensions. In *Proceedings of NAACL-HLT*, pages 540–545, 2016.

[300] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xue-jie Zhang. Predicting valence-arousal ratings of words using a weighted graph method. In *ACL (2)*, pages 788–793, 2015.

[301] Guido Zarrella, John Henderson, Elizabeth M Merkhofer, and Laura Strickhart. Mitre: Seven systems for semantic similarity in tweets. *Proceedings of the 9th International Workshop on Semantic Evaluation*, page 1217, 2015.

[302] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th CIKM*, pages 403–410, 2001.

[303] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, 2001.

[304] Jie Zhang, Yuan Wang, and Julita Vassileva. SocConnect: A personalized social network aggregator and recommender. *Information Processing & Management*, 49(3):721–737, 2013.

[305] Lei Zhang and Bing Liu. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 575–580, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[306] Jian Zhao, Liang Gou, Fei Wang, and Michelle Zhou. Pearl: An interactive visual analytic tool for understanding personal emotion style derived from social media. In *IEEE Conference on Visual Analytics Science and Technology*, pages 203–212. IEEE, 2014.

[307] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and traditional media using topic models. In *European Conference on Information Retrieval*, pages 338–349. Springer, 2011.

[308] Xin Zhao and Jing Jiang. An empirical comparison of topics in Twitter and traditional media. *Singapore Management University School of Information Systems Technical paper series. Retrieved November*, 10:2011, 2011.

[309] Xiang Zhu, Jiuming Huang, Bin Zhou, Aiping Li, and Yan Jia. Real-time personalized twitter search based on semantic expansion and quality model. *Neurocomputing*, 2017.

[310] Xiang Zhu, Jiuming Huang, Sheng Zhu, Ming Chen, Chenlu Zhang, Li Zhenzhen, Huang Dongchuan, Zhao Chengliang, Aiping Li, and Yan Jia. NUDTSNA at TREC 2015 microblog track: A live retrieval system framework for social network based on semantic expansion and quality model. 2015.

[311] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine learning*, 3(1):1–130, 2009.

Appendix A

Screening Questionnaire

Identification number: _____

1. At what level do you think your understanding of written English is?
   - Excellent
   - Very good
   - Good
   - Acceptable
   - Bad
   - Very bad
   - None
2. How familiar are you with interactive user interfaces such as dragging objects from one place to another?
   - Very familiar
   - Familiar
   - Very little
   - Not familiar
   - Not familiar at all
3. What is the highest level of education you have completed?
   - Little or no formal education
   - High school or equivalent
   - College or university
   - Master
   - Doctoral
   - Post-Doctoral
4. Are you color-blind?
   - Yes
   - No
5. What is your primary area of study?
   - Computer Science
   - E-commerce
   - Other_____

Appendix B

Demographic Questionnaire

Identification number: _____
Gender:_____Male _____Female
Age:_____

1. How long has it been since you last used a computer?
    - Less than 1 day
    - 1 day to less than 1 week
    - 1 week to less than 1 month
    - 1 month to less than 6 months
    - 1 months to less than 1 year
    - More than 1 year
2. On the average, how much time do you spend per week on a computer?
    - Less than 1 hour
    - 1 to less than 4 hours
    - 4 to less than 10 hours
    - 10 to less than 20 hours
    - 20 to less than 40 hours
    - Over 40 hours
3. How often do you use an interactive user interfaces such as dragging objects from one place to another?
    - Extremely often
    - Very often
    - Often
    - Not often
    - Seldom
    - Never
4. How comfortable are you at using interactive user interface?
    - Extremely comfortable
    - Very comfortable
    - Comfortable

- Uncomfortable
- Very uncomfortable
- Extremely uncomfortable

5. How often do you shop online?

   - Every day
   - Once every two days
   - Once every four days
   - Once a week
   - Once a month
   - Once a year
   - Never

6. How often do you read online product reviews?

   - Every day
   - Once every two days
   - Once every four days
   - Once a week
   - Once a month
   - Once a year
   - Never

Appendix C

Post-study Questionnaire

Identification number: _____

1. How well do you know the general topic of sentiment analysis?
   _____Very well _____Well _____Neutral _____Not well _____Not well at all

2. How easily did you assign sentiment values?
   _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

3. How confident are you of the sentiment values you assigned to sentiment pairs?
   _____Very confident _____Confident _____Neutral _____Not confident _____Not confident at all

4. How easy is the text-based interface to use?
   _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

5. How easy is the visual-based interface to use?
   _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

6. How helpful is the whole text-based interface in assigning sentiment values?
   _____Very helpful _____Helpful _____Neutral _____Not helpful _____Not helpful at all

7. How helpful is the whole visual interface in assigning sentiment values?
   _____Very helpful _____Helpful _____Neutral _____Not helpful _____Not helpful at all

8. Overall, which user interface do you prefer?
   _____Text interface _____Visual interface

9. How easily did you find different categories of aspects in the tree cloud?
   underline Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

10. How easy is the identification of un-seen branches?
    _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

11. How useful is the "tree cloud" in making the task easier?
    _____Very useful _____Useful _____Neutral _____Not useful _____Not useful at all

12. How easy is the navigation between views in the visual-based interface?
    _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

13. How easy is the polarity assignment in the visual-based interface?
    _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

14. How easy is the identification of sentiment words in the visual-based interface?
    _____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

15. How easy is the separation of sentiment pair(s) with the same sentiment word in the "polarity assignment view", i.e. clicking on a tag and duplicating the sentiment word?
_____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

16. How easy is to understand the sentiment value encoding, i.e. using color to show the sentiment value?
_____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

17. How easy is to understand the frequency encoding, i.e. using font-size to show the frequency?
_____Very easy _____Easy _____Neutral _____Difficult _____Very Difficult

18. How well suited are the visualization components for the operations?
_____Very well _____Well _____Neutral _____Not well _____Not well at all

19. How appealing did you find the design and layout of the visual components?
_____Very appealing _____Appealing _____Neutral _____Not appealing _____Not appealing at all

20. With proper documentation, how well do you think you could use the interface in the future?
_____Very sure _____Sure _____Neutral _____Not sure _____Not sure at all

21. Please give us more comments about the system:

_____
_____
_____
_____
_____

22. Are there any operations or visual components you expect to be included but were not available?

_____
_____
_____
_____
_____

Appendix D

Dalhousie Ethic Board's Letter of Approval

Subject	**REB # 2013-2921 Letter of Approval**
From	<sharon.gomes@dal.ca>,
To	<niri@cs.dal.ca>,
Cc	<eem@cs.dal.ca>, <sbrooks@cs.dal.ca>, <sharon.gomes@dal.ca>,
Date	2013-02-26 08:35

roundcube

DALHOUSIE UNIVERSITY
Inspiring Minds

Social Sciences & Humanities Research Ethics Board
Letter of Approval

February 26, 2013

Ms Raheleh Makki Niri
Computer Science\Computer Science

Dear Raheleh Makki,

REB #: 2013-2921
Project Title: Context-Specific Sentiment Lexicon Expansion Via Minimal User Interaction

Effective Date: February 26, 2013
Expiry Date: February 26, 2014

The Social Sciences & Humanities Research Ethics Board has reviewed your application for research involving humans and found the proposed research to be in accordance with the Tri-Council Policy Statement on *Ethical Conduct for Research Involving Humans*. This approval will be in effect for 12 months as indicated above. This approval is subject to the conditions listed below which constitute your on-going responsibilities with respect to the ethical conduct of this research.

Sincerely,

[redacted]

Dr. Sophie Jacques, Chair

Post REB Approval: On-going Responsibilities of Researchers

After receiving ethical approval for the conduct of research involving humans, there are several ongoing responsibilities that researchers must meet to remain in compliance with University and Tri-Council policies.

1. Additional Research Ethics approval
Prior to conducting any research, researchers must ensure that all required research ethics approvals are secured (in addition to this one). This includes, but is not limited to, securing appropriate research ethics approvals from: other institutions with whom the PI is affiliated; the research institutions of research team members; the institution at which participants may be recruited or from which data may be collected; organizations or groups (e.g. school boards, Aboriginal communities, correctional services, long-term care facilities, service agencies and community groups) and from any other responsible review body or bodies at the research site

2. Reporting adverse events
Any significant adverse events experienced by research participants must be reported **in writing** to Research Ethics **within 24 hours** of their occurrence. Examples of what might be considered "significant" include: an emotional breakdown of a participant during an interview, a negative physical reaction by a participant (e.g. fainting, nausea, unexpected pain, allergic reaction), report by a participant of some sort of negative repercussion from their participation (e.g. reaction of spouse or employer) or complaint by a participant with respect to their participation. The above list is indicative but not all-

inclusive. The written report must include details of the adverse event and actions taken by the researcher in response to the incident.

3. Seeking approval for protocol / consent form changes
Prior to implementing any changes to your research plan, whether to the protocol or consent form, researchers must submit them to the Research Ethics Board for review and approval. This is done by completing a Request for Ethics Approval of Amendment to an Approved Project form (available on the website) and submitting three copies of the form and any documents related to the change.

4. Submitting annual reports
Ethics approvals are valid for up to 12 months. Prior to the end of the project's approval deadline, the researcher must complete an Annual Report (available on the website) and return it to Research Ethics for review and approval before the approval end date in order to prevent a lapse of ethics approval for the research. Researchers should note that no research involving humans may be conducted in the absence of a valid ethical approval and that allowing REB approval to lapse is a violation of University policy, inconsistent with the TCPS (article 6.14) and may result in suspension of research and research funding, as required by the funding agency.

5. Submitting final reports
When the researcher is confident that no further data collection or analysis will be required, a Final Report (available on the website) must be submitted to Research Ethics. This often happens at the time when a manuscript is submitted for publication or a thesis is submitted for defence. After review and approval of the Final Report, the Research Ethics file will be closed.

6. Retaining records in a secure manner
Researchers must ensure that both during and after the research project, data is securely retained and/or disposed of in such a manner as to comply with confidentiality provisions specified in the protocol and consent forms. This may involve destruction of the data, or continued arrangements for secure storage. Casual storage of old data is not acceptable.

It is the Principal Investigator's responsibility to keep a copy of the REB approval letters. This can be important to demonstrate that research was undertaken with Board approval, which can be a requirement to publish (and is required by the Faculty of Graduate Studies if you are using this research for your thesis).

Please note that the University will securely store your REB project file for 5 years after the study closure date at which point the file records may be permanently destroyed.

7. Current contact information and university affiliation
The Principal Investigator must inform the Research Ethics office of any changes to contact information for the PI (and supervisor, if appropriate), especially the electronic mail address, for the duration of the REB approval. The PI must inform Research Ethics if there is a termination or interruption of his or her affiliation with Dalhousie University.

8. Legal Counsel
The Principal Investigator agrees to comply with all legislative and regulatory requirements that apply to the project. The Principal Investigator agrees to notify the University Legal Counsel office in the event that he or she receives a notice of non-compliance, complaint or other proceeding relating to such requirements.

9. Supervision of students
Faculty must ensure that students conducting research under their supervision are aware of their responsibilities as described above, and have adequate support to conduct their research in a safe and ethical manner.

Appendix E

Tweet Fields

The list of tweet fields[1] that are used in this thesis with their name in the Twitter API and their description.

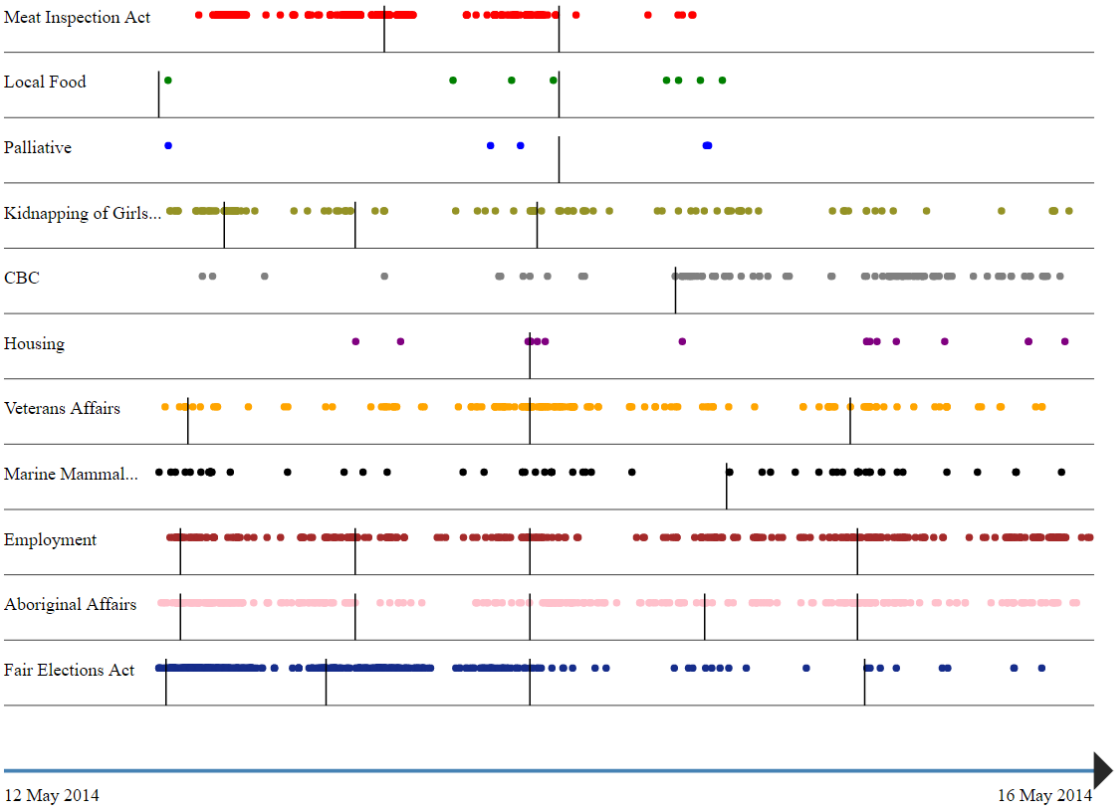
Terminology	Field Name	Description
id	id	a unique identifier for the tweet
text/content	text	the UTF-8 text of the tweet (status update)
date/time	created_at	the time (UTC) that the tweet was published
hashtags	hashtags	a word or phrase prefixed with the symbol '#'. The field contains a list of hashtags that appear in the text of the tweet
urls	urls	list of URLs that appear in the text of the tweet
user_mentions	user_mentions	list of other users' names preceded by the symbol '@', that are mentioned in the text of the tweet
retweet	retweeted_status	retweet is broadcasting of a tweet that has been authored by other users. The field is a representation of the original tweet that was retweeted, and its existence indicates that the tweet is a retweet
reply	in_reply_to_status_id	a tweet that replies to an original tweet. The field contains the id of the original tweet, and if its value is not null, it means that the tweet is a reply tweet
user	user:id	anyone or anything with an account
author	user:id	the user who published the tweet
number of user's followers	followers_count	number of accounts that are following the user
number of user's followees	friends_count	number of accounts that the user is following
number of user's tweets	statuses_count	the number of tweets (including retweets) the user has posted

[1]For the complete list of fields in a tweet, visit the Twitter website `https://dev.twitter.com/overview/api/tweets`.

Appendix F

Tweet Distribution Over Time

Figure F.1: Distribution of relevant tweets over time, each vertical line shows the time when each debate was discussed in the parliament.



Appendix G

Evaluation Metrics

TP_{d_j} = number of true positive tweets for debate d_j

FN_{d_j} = number of false negative tweets for debate d_j

TN_{d_j} = number of true negative tweets for debate d_j

FP_{d_j} = number of false positive tweets for debate d_j

M = number of classes (debates)

$accuracy = \frac{\sum_{j=1}^{M}(TP_{d_j}+TN_{d_j})}{\sum_{j=1}^{M}(TP_{d_j}+FN_{d_j}+TN_{d_j}+FP_{d_j})}$

$precision(d_j) = \frac{TP_{d_j}}{TP_{d_j}+FP_{d_j}}$

$recall(d_j) = \frac{TP_{d_j}}{TP_{d_j}+FN_{d_j}}$

$macro - recall = \frac{\sum_{j=1}^{M} recall(d_j)}{M}$

$macro - precision = \frac{\sum_{j=1}^{M} precision(d_j)}{M}$

R = ranked list of retrieved documents

$p(k)$ = precision at rank k in the ranked list

$rel(k) = 1$ if document at position k is a true positive (relevant) document, otherwise $rel(k) = 0$

$Average\_Precision(d_j) = AP(d_j) = \frac{\sum_{k=1}^{|R|}(p(k)\times rel(k))}{TP_{d_j}+FN_{d_j}}$

$Mean\_Average\_Precision = MAP = \frac{\sum_{k=1}^{M} AP(d_i)}{M}$

$R - precision(d_j) = p(TP_{d_j} + FN_{d_j})$

Appendix H

Applying SVMRank for Tweet Filtering

$Q = \bigcup_{q_i \in \widetilde{Q}} q_i$ ▷ Q includes all initial query terms

$l = retrieval(Q)$ ▷ ranked list of tweets returned by the model in Subsection "Tweet Retrieval"

$TS = \bigcup_{d' < d} TS_{d'}$ ▷ add all labeled tweets from previous days to the training set, i.e. TS

$TS_d = 0$ ▷ initialize the training set for day d

while $size(TS_d) < limit$ **do** ▷ limit of the number of labeling requests

  if $l \neq \emptyset$ **then**

    $t_j = l.head()$ ▷ remove and get top tweet

    if $\forall t_{j'} \in TS, near\_dup(t_j, t_{j'}) == false$ **then** ▷ Subsection "Novelty Verification"

      $lbl = label(t_j)$ ▷ ask the information source to label t_j

      $TS_d = TS_d \cup (t_j, lbl)$

      $TS = TS \cup (t_j, lbl)$ ▷ add labeled tweet t_j to the training set

if $\nexists t_j \in TS$, that $lbl(t_j) == not\ relevant$ **then**

  $t_j = l.tail()$ ▷ remove and get tweet at the end of the ranked list

  $TS = TS \cup (t_j, not\ relevant)$ ▷ assume tweets t_j is not relevant and add it to TS

if $\nexists t_j \in TS_d$, that $lbl(t_j) == relevant$ **then** ▷ if there are no relevant tweets in TS_d

  return $R_d = \emptyset$ ▷ remain silent for day d

else

  $trainSVMRank(TS)$ ▷ train SVMRank

  $R_d = testSVMRank(C_d)$ ▷ ranked list of relevant tweets after applying SVMRank

  $R_d = verifyNovelty(R_d)$ ▷ return R_d after applying "Novelty Verification"

Appendix I

Applying RoundRobin to UTIF

1: $Q = \bigcup_{q_i \in \widetilde{Q}} q_i$ ▷ Q includes all initial query terms
2: $l = UTIF(Q)$ ▷ ranked list of tweets returned by UTIF
3: $counter = 0$ ▷ keeping the number of asked labeling requests
4: **while** $counter < limit$ **do** ▷ $counter <$ limit of the number of labeling requests
5: $\quad rel = \emptyset$ ▷ rel contains labeled tweets that are relevant
6: $\quad irrel = \emptyset$ ▷ $irrel$ contains labeled tweets that are not relevant
7: $\quad l' = l$ ▷ create a copy of l
8: $\quad$ **if** $l' \neq \emptyset$ **then**
9: $\quad\quad t_j = l'.head()$ ▷ remove and get top tweet
10: $\quad\quad lbl_j = label(t_j)$ ▷ ask the information source to label t_j
11: $\quad\quad$ **if** $lbl_j == relevant$ **then**
12: $\quad\quad\quad rel = rel \cup (t_j, lbl_j)$ ▷ add relevant tweet to rel
13: $\quad\quad$ **else**
14: $\quad\quad\quad irrel = irrel \cup t_j$ ▷ add irrelevant tweet to $irrel$
15: $R_d = sort(rel)$ ▷ output list of tweets is initialized to the sorted list of relevant tweets
16: **while** $l \neq \emptyset$ **do**
17: $\quad t_j = l.head()$
18: $\quad$ **if** $(t_j \notin R_d) \wedge (t_j \notin irrel)$ **then**
19: $\quad\quad$ add t_j to R_d

Appendix J

Performance of Tweet Filtering Systems

Table J.1: Profiles and their nDCG-0@10 value when UTIF, ACTIF and SVMRank are applied considering 1020 labeling requests

Profile	Title	Rel.[a]	NER	Sil.[b]	UTIF	ACTIF	SVMRank
MB226	Hershey PA quilt Show	0	✓	10	0.0000	0.0000	0.0000
MB227	Pradaxa side effects	0		10	0.0000	0.0000	0.0000
MB228	Coumadin dietary restrictions	3		7	0.0000	0.0000	0.0000
MB236	California drought agricultural effects	463	✓	0	0.3793	**0.4561**	0.1608
MB242	Saudi bombing Yemen	181	✓	1	0.0812	**0.3451**	0.1151
MB243	FIFA corruption investigation	466	✓	0	0.3454	**0.3752**	0.2284
MB246	Greek international debt crisis	422	✓	0	0.3014	**0.4623**	0.2071
MB248	Harlem 5K race	0	✓	10	0.0000	0.0000	0.0000
MB249	John Hopkins Lyme disease study	0		10	0.0000	0.0000	0.0000
MB253	Health insurance for disabled children	3		7	0.0000	**0.2000**	**0.2000**
MB254	Cancer and depression	41		3	0.0613	**0.2007**	0.1136
MB255	Medical insurance on cruises	18		1	0.0000	0.0469	**0.0765**
MB260	Society for Women and the Civil War Conference	8	✓	4	0.0000	0.0000	0.0000
MB262	Stephen Colbert Late Show	126	✓	0	0.1161	**0.5659**	0.4052
MB265	Cruise ship mishaps	101		0	0.0000	**0.2196**	0.0853
MB267	Fighting between Ukraine and pro-Russian rebels	80	✓	0	0.1209	**0.1639**	0.1196
MB278	Mr. Holmes movie	44	✓	1	0.5643	**0.6154**	0.3998
MB284	Coping with identity theft	176		0	0.0000	**0.2142**	0.1796
MB287	The Vatican Tapes movie	109		0	0.0399	**0.6165**	0.5721
MB298	Gaza rockets hit Israel	9	✓	4	0.2613	**0.3406**	0.2613
MB305	National Museum of American History	4	✓	8	0.0000	0.0000	0.0000
MB324	Indian-Pacific train	3	✓	8	0.0000	0.0000	0.0000
MB326	Wheelchair accessibility	66		0	0.1079	**0.2073**	0.0823
MB331	Special Olympics 2015	198	✓	0	0.0707	0.1385	**0.1543**
MB339	Chincoteague Pony Swim	14		5	0.2190	0.2190	**0.4209**
MB344	Iran nuclear agreement	1707	✓	0	**0.7748**	0.5718	0.6252
MB348	Drones vs. commercial airliners	116		1	0.0000	**0.1220**	0.0220
MB353	Summer Seasonal Affective Disorder SAD	13		2	0.0000	**0.0793**	0.0000
MB354	Go Set a Watchman	143	✓	0	**0.1114**	**0.1114**	0.0691
MB357	Prevalence of Ritalin use with no ADHD diagnosis	7		5	0.0000	0.0000	**0.0469**
MB359	Grey book	9	✓	4	**0.1343**	**0.1343**	0.0282
MB362	Outback Steakhouse	67	✓	0	0.3756	0.3756	**0.3881**
MB366	Climbing Mount Everest	152	✓	0	0.1420	**0.3557**	0.2688
MB371	Self-driving cars	290		0	**0.5251**	0.5088	0.3684
MB377	Animal attacks in safari parks	26		2	0.0000	**0.2073**	0.0469
MB379	Morel mushrooms	52		0	0.0693	0.0693	**0.3761**
MB383	Online dating for older women	21		2	0.0000	**0.0920**	0.0000
MB384	Arson fires in inner cities	18		3	0.0613	**0.2916**	0.2107
MB389	Clinton Foundation	21	✓	1	0.6273	**0.7043**	0.5738
MB391	Polar icecap melting	135		0	0.0000	**0.3079**	0.0979
MB392	U.S. forest fires	80	✓	0	0.0549	**0.1174**	0.0363
MB400	Probiotics	109		0	0.5351	0.5351	**0.7010**
MB401	Knock Knock Live	2042		0	**0.4436**	0.3960	0.2685
MB405	Rotterdam Unlimited	5	✓	7	0.0613	**0.1411**	0.1226
MB409	Airport TSA screenings	39	✓	0	**0.0282**	**0.0282**	**0.0282**
MB416	Hepworth Exhibit at the Tate Britain	13	✓	3	0.0000	0.2793	**0.3793**
MB419	King George Weekend Ascot	118	✓	1	0.0290	0.2766	**0.4000**
MB432	Mount Rushmore	50	✓	0	0.3763	**0.5657**	0.3796
MB434	2015 Summer PanAm Games	373	✓	1	0.0366	**0.2533**	0.1577
MB439	Bolton Wanderers	86	✓	0	0.2628	0.2628	**0.2715**
MB448	Bouchercon World Mystery Convention	6		5	0.0000	**0.1000**	**0.1000**

[a] Number of relevant tweets in the subset of judged tweets

[b] Number of silent days in the evaluation period (July 20-29)