Exploiting Temporal and Semantic Information for Microblog Retrieval through Query Expansion and Reranking Approaches

(クエリの拡張と再順位付けアプローチによるマイクロブログ検索のための時間的および意味的情報の活用)

January, 2019

Doctor of Philosophy (Engineering)

Abu Nowshed Chy

アブ ノウシャド チョウドリ

Toyohashi University of Technology

I would like to dedicate this thesis
To my beloved
Father Abu Hena Chy,
and
Mother Kowser Parbin.

Acknowledgements

At first, I want to express my gratitude to the Almighty for His endless kindness for keeping me mentally and physically fit to complete this sophisticated task.

This study would not get its own shape without the special support of Japan's Ministry of Education, Culture, Sports, Science, and Technology (MEXT) who provided me with the funding throughout my study period in Japan. Though the following dissertation is an individual work, it could never have reached the heights without the help, support, guidance, and efforts of so many peoples in so many ways.

My supervisor, Prof. Masaki Aono, has been a significant presence in my life. His continuous supervision during my Master's and Doctoral program was a true gift and his insights have strengthened this study significantly. I will always be thankful to him for his kind support. His directions and guidelines of research, preparing academic manuscripts, reports, presentations, and posters make me more dynamic and well-organized. I do not hesitate to certify him as the best academic scholar so far I have experienced. I also thankful to our former Assistant Prof. Atsushi Tatsuma for providing me good comments and suggestions during laboratory seminar presentations.

Besides my supervisor, I would like to thank the rest of my thesis committee: Prof. Kyoji Umemura and Associate Prof. Tomoyosi Akiba for their insightful comments and suggestions several times. Moreover, I take this opportunity to express my gratitude to all the faculty members especially Prof. Shigeki Nakauchi, Prof. Shigeru Masuyama, Prof. Toshihiro Fujito, Prof. Shigeru Kuriyama, Prof. Yoshiteru Ishida, and Dr. Kazuho Watanabe for their comments and

encouragements, which incented me to refurbish my research from various perspectives. I also give thanks to all the office staff at TUT especially staff at the International Affairs Division (IAD) for their continuous support.

I am especially grateful to former lab member Md Zia Ullah for his tremendous supports and guidelines to complete my research work. I have been extremely lucky to have a senior lab member like him who cared so much about my work and who responded to my every questions and queries so promptly. I also want to recollect all of my labmates who always inspire, help and motivate me during this long journey. Today, I wish to remember all of my seniors especially Ismat Ara Reshma, Nihad Karim Chowdhury, Vanna-san, Sakurada-san, Koyanagi-san, and others. My heartiest thanks also go to other lab members for their supportive mentality and help namely Md Shajalal-san, Tashiro-san, Watanabe-san, Yoshii-san, Hamada-san, Himeno-san, Endo-san, Takashima-san, and others.

My special thanks go to Prof. Md. Hanif Seddiqui who recommended and introduced me to my honorable supervisor. Moreover, I would like to thanks Rudra Pratap Debnath for his special support during my MEXT application process.

I also grateful to my family members and relatives, especially to my father Abu Hena Chy, mother Kowser Parbin, wife Umme Aymun Siddiqua, elder sister Tanjila Nowshin, brother-in-law Sultan Uddin Ahmed, and younger brother Abu Nowhash Chy. Their patience, love, and encouragement have upheld me throughout my journey in Japan. I also thank to all the persons of Bangladeshi community living in Toyohashi, Japan.

Finally, I would like to show my gratitude to the cultures, traditions, and Japanese mentality. I learn too many things from here.

Abu Nowshed Chy
January 31, 2019

Abstract

Nowadays, microblog websites are not only the places in maintaining social relationships but also act as a valuable information source. Everyday lots of users turn into these sites to fulfill their diverse information needs. Moreover, during a disaster period, microblogs are treated as an important source to serve the situational information needs. Among several microblog services, twitter is now the most popular. Hence, information retrieval in twitter has made a hit with a lot of complaisance.

However, due to the short length characteristics of tweets, people usually use unconventional abbreviations (e.g. use "2day" instead of "today"), poor linguistic phrases (e.g. use "TYT" instead of "take your time"), and URL to express their concise thought. Besides this, some twitter specific syntaxes (e.g. #hashtags, retweets) also very popular among twitter users. Moreover, users usually search the temporally relevant information in twitter, such as breaking news and real-time events. All these characteristics make it challenging for effective information retrieval (IR) over tweets.

In this thesis, we propose two different approaches to tackle the challenges of microblog retrieval. At first, we propose a reranking based approach, where the main focus is to rerank the tweets that are retrieved by using the baseline retrieval model. Whereas, our proposed query expansion based approach augments the original queries with expansion terms that best represent the users' intent. In both approaches, we used the Lucene's implementation of query likelihood as the baseline retrieval model.

In our reranking based approach, we emphasize the alleviation of vocabulary mismatch, and the leverage of the temporal (e.g. recency and burst nature) and contextual characteristics of tweets. One way of alleviating the vocabulary mismatch problem is to reformulate the query via query expansion. In this regard, we propose a three-stage query expansion technique by leveraging the pseudo-relevant tweets at the first stage, made use of Web search results at the second stage, and extracted hashtags relevant to the query at the third stage. For weighting terms, we used the IDF-score of each term.

In the feature extraction stage, we extract several effective features for reranking by leveraging the different tweet characteristics. To extract temporal aspect of tweets, we determine the temporal dimension of the query and if the query is temporally sensitive, we extract our proposed temporal features such as recency score of tweets by utilizing the query time and tweet time. We also hypothesize that any tweet might have burst-time popularity once the tweet has been posted. In order to implement our hypothesis, we propose to introduce a "burst-time" aware temporal feature as well. To extract the content-aware features, we utilize some classical information retrieval models. Along with this direction, we extract the twitter specific features such as URL and retweet count and account related features such as followers count and status count to address the special characteristics of tweets and relations between twitter users. Moreover, we propose some context relevance features based on word embedding, kernel density estimation, and query-tweet sentiment correlation to address the contextual dimension of tweets. We hypothesize that a query is sentimentally sensitive if the largest proportion of the initially retrieved tweets has the similar kind of sentiment polarity. Once our proposed features are extracted, a supervised feature selection method based on regularized regression is applied to select the best feature combination. After estimating the feature importance using the random forest, an ensemble of learning to rank (L2R) framework is applied to estimate the relevance of a query-tweet pair.

However, the naive query expansion technique that we proposed in our reranker framework used the IDF-score to rank each term which might induce irrelevant rare terms from the noisy tweet contexts. Hence, selecting terms by utilizing the unsupervised approach and highly reliant on the top retrieved results without considering temporal relevance may generate the noisy or harmful expansion terms which degrade the retrieval performance.

Considering the above limitations, we present another query expansion approach, where supervised learning is adopted for selecting expansion terms. Upon retrieving tweets by our proposed topic modeling based query expansion (TMQE), we utilize the pseudo relevance feedback (PRF) and a new temporal relatedness (TR) approach to select the candidate tweets. Next, we devise several new features to select the temporally and semantically relevant expansion terms by leveraging the temporal, word embedding, and sentiment association of candidate term and query. Moreover, we also utilize the lexical and twitter specific features to quantify the term relatedness. After supervised feature selection using regularized regression, we estimate the feature importance by applying random forest. Then, we design a linear learning to rank (L2R) model with the aid of feature values and their importance weight to rank the candidate expansion terms.

Experimental results on TREC Microblog 2011 and 2012 test collections over the TREC Tweets2011 corpus demonstrate the effectiveness of our proposed reranking and query expansion approaches over the baseline and state-of-the-art methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The rapid growth of microblog platforms such as twitter, tumblr, sina weibo, etc. provides a convenient way to the users for sharing their views, experiences, opinions, breaking news, and ideas as well as interacting with others anytime, from anywhere. Everyday lots of users turn into these microblog sites to get some information what is happening around the world as well as fulfill their diverse information needs. That is why individuals and organizations are increasingly seeking ways to analyze the peoples' opinionated data generated in these media due to its wide range of applications across numerous platforms including market research, business intelligence, enhancement of online shopping infrastructures, predicting the stock market, political polls, scientific surveys from a psychological and sociological perspective, and so on.

Moreover, due to the real-time nature of the microblog sites, these sites are treated as an important source to serve the situational information needs during a disaster period [4]. Monitoring and analyzing massive microblog posts to produce the curated contents based on different information types provide enormous opportunities to different public safety personnel for reducing casualties, preventing secondary disasters, economic losses, social disruption etc. [5] as well as post-incident analysis.

Among several microblog sites, Twitter[1] is now the most popular, where lots of users post tweets whenever a notable event occurs. With the rapid growth of the internet, the number of twitter user increases from the year 2006 to till now. It is therefore necessary to think about:

- Why the popularity of twitter increases?
- What attracts people to share information via twitter posts?
- What kind of information people share in twitter?
- What information people usually search in twitter?
- How twitter posts evolved with time?
- What kind of factor makes it challenging for designing effective information retrieval (IR) system over tweets.

For addressing such questions and boosting the retrieval effectiveness, text retrieval conference (TREC) was first introduced the microblog ad-hoc search task in 2011 [6], where a user's information need had been represented by a query at a specific point in time and a set of relevant ranked tweets had been returned. After that information retrieval in twitter has made a hit with a lot of complaisance.

However, queries provided by users are usually too short, ambiguous, and hardly describe the information need accurately. For example, by issuing the query "Buying Clothes Online", a searcher might look for online stores, coupons, discussions, suggestions, or even remote try-on technologies that are related to online shopping [7]. Since tweets have a length constraint, people usually use unconventional abbreviations, poor linguistic phrases, URL as well as other twitter specific syntaxes (e.g. retweet, #hashtag) to express their concise thought. Moreover, searching tweets on Twitter, users seek information with temporal relevance in mind, such as breaking news and real-time events [8]. For example, when the breakup news of famous band "White Stripes" published on 2nd Feb, 2011, many people post tweets about this topic on that day. Therefore, posts that are generated before or after this date are less relevant to the query, "White Stripes breakup." We will discuss some other challenges in the next segment that make the ad-hoc search in microblog environment a formidable task.

[1] https://twitter.com

1.2 Research Challenges in Microblogosphere

To improve the performance of microblog retrieval especially in twitter, we need to address several challenges including: vocabulary mismatch problem, temporal relevance, short length of tweet text, social attributes, retweets, low quality of tweet text, contextual dimension, different twitter specific characteristics that makes the microblog search different from traditional web search, and so on. Now, we briefly describe these challenges:

- **Vocabulary Mismatch Problem:** Tweet texts have different characteristics from the traditional web document. Moreover, due to the length constraint of tweets people usually use informal texts that filled with accidental and deliberate spelling errors. All these characteristics cause the severe vocabulary mismatch problem which exacerbates the difficulty of query-tweet matching during retrieval.

- **Temporal Relevance:** People usually posts in twitter to share the current events information, breaking news, experiences as well as the opinion towards various target entities. An important characteristic of twitter is that people tend to post about a topic within a specific period of time (i.e. bursty nature). Since people search microblog posts with temporal relevance in mind [8], temporal information impacts the performance of the retrieval model. Hence, it is important to address the temporal aspects of microblogs for designing effective retrieval model.

- **Short Length of Microblog Text:** Tweets have a length constraint (previously limited to 140 characters in length and currently 280 characters in length), which restrict the people to express their thoughts concisely. Therefore it is challenging to distill the concise information for estimating the relevance of query-tweet pair.

- **Low Quality of Tweets:** The short length constraint of the tweet makes characters expensive. To overcome the restricted length, people usually use informal and ungrammatical texts containing lots of emoticons, symbols,

slangs, spelling errors, etc. in their tweets. Extracting information from such noisy tweet texts is challenging.

- **Social Attributes like #hashtag:** People are trying to introduce new ways to express their thought concisely to overcome the length constraint of twitter. A twitter #hashtag is a type of label or metadata tag preceded by a '#' sign used by users within a tweet to highlight the trending events or issues. Similar kind of hashtags indicates a similar context. Segmenting the hashtags and extracting information from it might be important for better understanding the tweet texts.

- **Retweets/Near Duplicate Tweets:** Twitter allows the user to forward a tweet with or without modification to the followers of his/her account. This feature is called the retweet in Twitter. While retweeting can simply be seen as the act of copying and rebroadcasting, but it has an immense contribution in a conversational ecology. For this reason, some of the most visible twitter participants retweet others and intend to be retweeted [9]. Moreover, if a tweet retweeted lots of times it might contain some importance, which needs to be exploited in the tweet ranking model.

- **Contextual Dimension:** The brevity of tweets poses new challenges to the traditional IR techniques including ranking, classification, etc. Tweet texts do not provide sufficient statistical information for estimating the relevance score effectively and abbreviations as well as new words are used by the users incessantly. These problems also exacerbate the problems of synonymy (distinct words of the same meaning) and polysemy (same word with different meanings). Therefore it is important to extract the contextual dimension of the tweets for designing effective IR models in microblog domain [10].

- **Different from Traditional Web Search:** The real-time nature of the twitter makes it popular than the traditional web platform. People are increasingly turning into twitter to get the current updated news/information in a real-time manner, while it takes some time for the web platform to develop and learn about a topic.

1.3 Research Questions and Focus

The thesis focuses on the ad-hoc search task in microblog environment, such as twitter, which is one of the state-of-the-art research in information retrieval (IR) domain. To tackle this task, we employ two different approaches in our proposed framework: (1) reranking based approach and (2) query expansion based approach as shown in Figure 1.1.

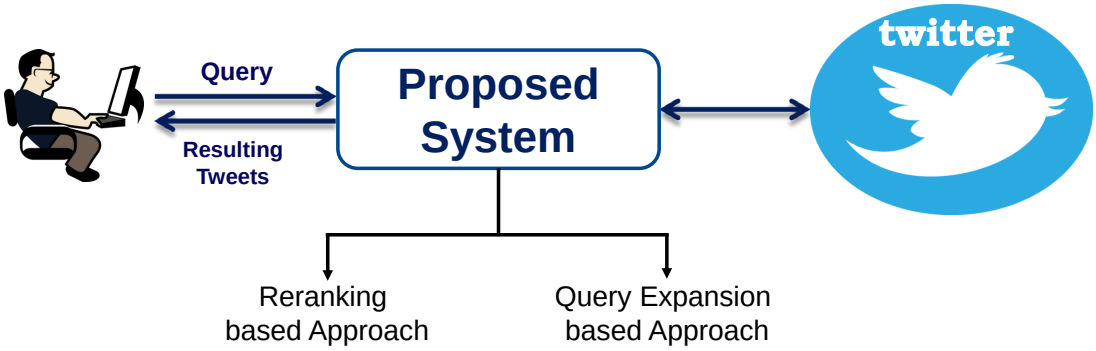

Figure 1.1: Overview diagram of our proposed microblog retrieval framework.

In the reranking based approach, given a query, our proposed method rerank the tweets retrieved by using the baseline retrieval model. Whereas, in the query expansion based approach, our proposed framework augment the original queries with expansion terms that best represent the users' intent through reducing the ambiguity of the query's original representation.

There is a long thread of research employing the reranking approach for microblog retrieval. Traditional retrieval models (e.g. Okapi BM25, language model, etc.), temporal property of the tweets as well as other twitter specific characteristics exploited as the feature by researchers [11, 12, 13, 14]. But a method that effectively exploits the temporal property and other twitter specific characteristics in a single framework absent in the previous study. On the other hand, a number of prior research applying the query expansion (QE) to enhance the performance of microblog retrieval [15, 16, 17, 18, 19]. Most of these methods utilize the pseudo-relevance feedback (PRF) where only top retrieved tweets are used to select the expansion terms. However, highly reliant on the top retrieved tweets and selecting the terms using unsupervised approach may generate noisy or harmful expansion terms, which in turn degrade the retrieval performance [20, 21].

We now summarize the research questions (RQ) in terms of the microblog reranking approach that we intend to answer in this dissertation:

RQ1.1. Is the feature based reranking approach effective for microblog retrieval?

RQ1.2. How can we tackle the severe vocabulary mismatch problem between the query-tweet pair?

RQ1.3. How can we tackle the real-time nature of the twitter in our retrieval framework?

RQ1.4. What kinds of features are effective for microblog reranking model?

RQ1.5. Microblog queries may have temporal (temporal or non-temporal) and sentiment (sentiment sensitive or insensitive) sensitivity. How can we determine such sensitivity and classify the queries? Is the effectiveness of the reranking model affected by such classification?

RQ1.6. How successful our method to rerank the retrieved tweets in compared to the various state-of-the-art methods?

We now summarize the research questions (RQ) in terms of the query expansion approach that we intend to answer in this dissertation:

RQ2.1. Is the query expansion approach improve the retrieval performance of microblog retrieval?

RQ2.2. How effectively query expansion approach solve the vocabulary mismatch problem?

RQ2.3. Can we effectively apply topic modeling to select the candidate expansion terms?

RQ2.4. Does exploit the temporal relatedness of the query-tweet pair effective for candidate tweet selection?

RQ2.5. How to select the effective expansion terms in microblog domain?

RQ2.6. What kinds of features are important to select the effective expansion terms?

RQ2.7. How can we tackle the temporal and semantic relevance between candidate terms and query?

RQ2.8. How successful our method to select the effective expansion terms in compared to the various state-of-the-art methods?

1.4 Contributions

This research focuses on addressing the challenges of microblog retrieval. Hence, we present some novel contributions to enhance the performance of microblog retrieval system. In this segment, we will summarize our key contributions, and map them to the related research questions as well as the subsequent Sections for a detailed explanation. For better understanding, we organize our contributions in two main categories: (1) contributions that are proposed to enhance the performance of the microblog reranker system; (2) contributions that are proposed to enhance the performance of the query expansion system for microblog retrieval.

C1. We propose a microblog reranking framework that estimates the relevance of query-tweet pair by exploiting ensemble of features broadly grouped into content relevance features, twitter specific features, account related features, context relevance features, popularity based features, and temporal features. Other contributions are enlisted below:
[*The contributions are related to RQ1.1. and RQ1.4., which will be discussed in Section* 4.2].

- C1.1 To overcome the limitations of the vocabulary mismatch problem, we introduce four context relevance features based on word-embedding and query-tweet sentiment correlation. In this context, we also introduce a simple but effective three-stage query expansion technique that leverage the pseudo-relevant tweets at the first stage, made use of Web search results at the second stage, and extracted hashtags relevant to the query at the third stage.
[*The contributions are related to RQ1.2. and RQ1.4., which will be discussed in Sections* 4.2.4.4 and 4.2.2].
- C1.2 Since user search information in twitter with temporal relevance in mind, we introduce two effective temporal features for addressing the temporal aspects (recency and temporal variations) of tweets.
[*The contributions are related to RQ1.3. and RQ1.5., which will be discussed in Section* 4.2.4.6].

C1.3 Queries issued by the users may have temporal or sentiment sensitivity which poses the greater influence on the retrieval effectiveness. To determine the queries temporal and sentiment sensitivity, we introduce a query type determination technique in our proposed framework.
[*The contributions are related to RQ1.5., which will be discussed in Section* 4.2.3].

C1.4 We introduce our own version of URL popularity and hashtag importance features to estimate the importance of tweets.
[*The contributions are related to RQ1.4., which will be discussed in Section* 4.2.4.5].

C2. We propose a query expansion (QE) framework for microblog retrieval that augments the query by selecting the effective expansion terms under the supervised manner. Other contributions are enlisted below:
[*The contributions are related to RQ2.1. and RQ2.2., which will be discussed in Section* 5.2].

C2.1 Tweets are modeled as a mixture of topics and topics underlying within tweets may be an important piece of information to distill its content. In our proposed query expansion framework, we introduce an effective topic modeling based query expansion (TMQE) technique to improve the baseline retrieval.
[*The contributions are related to RQ2.1., RQ2.2., and RQ2.3., which will be discussed in Section* 5.2.2].

C2.2 An important characteristic of twitter is that people are discussed about a topic within a specific period of time and tweets that are posted within this active temporal area might be relevant to this topic. Therefore, instead of obtaining expansion terms only from top-ranked tweets as does pseudo-relevance feedback (PRF), we introduce a temporal relatedness (TR) approach based on C-LSTM for candidate tweet selection which generates the pool of effective candidate terms.
[*The contributions are related to RQ2.4. and RQ2.5., which will be discussed in Section* 5.2.3.2].

C2.3 To bridge the temporal and semantic gaps between the candidate terms and query, we propose new temporal, sentiment aware, and word embedding based features by leveraging the temporal (temporal correlation and recency) and contextual aspects of the candidate terms.
[*The contributions are related to RQ2.6. and RQ2.7., which will be discussed in Sections* 5.2.5.3, 5.2.5.4, and 5.2.5.5].

Other contributions of this research include a thorough review of the literature in microblog domain as well as related IR techniques and improvements of our proposed approaches over the state-of-the-art methods.
[*The contributions are related to RQ1.6. and RQ2.8., which will be discussed in Chapter 3 and Section* 4.3.5 and 5.3.4].

1.5 Thesis Organization

The rest of the thesis is organized as follows:

— In **Chapter 2**, we present the background concepts that are related to our thesis. We refer readers, who are new in this field, to read this chapter to comprehend the consequent contents of this thesis.

— In **Chapter 3**, we present a details literature review about the state-of-the-art of microblog information retrieval.

— In **Chapter 4**, we focus on our proposed microblog reranker framework. It includes detailed scope and verification of our proposed reranking method as well as experiments and comparative evaluation with other related methods to show the effectiveness of our proposed method.

— In **Chapter 5**, we focus on our proposed query expansion framework for microblog retrieval. We present our experiments and findings, comparing with the other state-of-the-art methods to demonstrate the effectiveness of our method.

— In **Chapter 6**, we present the overall conclusions of our thesis and some plausible future directions.

— In **Appendix A**, we present our experiments related to microblog retrieval in a disaster situation, which is based on the TREC incident streams task.

Chapter 2

Background Concepts

2.1 Information Access

With the widespread online information sharing platforms, nowadays people are increasingly interested in information access and communication technologies. The term *information access (IA)* [22] refers to the findability of the information regardless of its format, channel, or location. This definition focuses on how find-able you make your information, how you emphasis on the success of your information management regimen, and how effectively you incorporate the user experiences into the search process in comparison to the state-of-the-art search algorithms.

According to the information analyst firm Gartner[1], information access is a collection of tools and technologies which objective is to help in finding the required information. The technologies include but is not limited to the following list:

- Information Search
- Content classification, categorization, and clustering
- Entity extraction
- Fact checking
- Taxonomy creation and management
- Visualization of information

[1] https://www.gartner.com/en

To look for any types of information people usually employ three main ways including:

- Pattern matching or search - given a information, the goal is to finding information that holds the same attributes such as contains similar words or phrases or words that are usually exists close to each other (e.g. clustering).
- Semantic web navigation or traversal - knowledge of a relevant information type to find the other information and traverse the links based on information relevancy.
- Classified or categorized - finding information by browsing the organized topic. Classification taxonomies and structured organizations of information are employed in this regard.

However, to access the information properly, first, we need to organize it properly. Various research efforts in information access are employed to access and process the large amounts of data and information effectively and comfortably [23]. These efforts are broadly covered by the general area of information retrieval, text mining, text classification, machine translation, etc.

2.1.1 Information Retrieval

According to the book "An Introduction to Information Retrieval" [24], information retrieval (IR) can be defined as follows:

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

The user's information need can be represented by a *query* or *profile* which may contain one or more *search terms* as well as some additional information. Hence, the information retrieval system retrieves the relevant information by comparing the terms of the query with the *indexed terms* appearing in the document itself. The decision may be binary such as relevant/non-relevant or estimating the degree of relatedness between the query-document pair. Information retrieval systems can also be distinguished by the scale at which they operate. For example, in *web search*, the system needs to provide search result generated from billions of web

documents that stored on millions of computers around the world. For indexing such a massive amount of data to provide effective search results the system need to consider several distinctive issues. Moreover, the system also needs to consider specific aspects of the web, such as the exploitation of web pages and also not being fooled by site providers because to boost up their rankings in the search result they usually try to manipulate their page contents with the search engine optimization (SEO) techniques. *Personal information retrieval* may also be considered at the other extreme. In recent years, information retrieval system is integrated with the consumer operating systems such as Apple's Mac OS X Spotlight, Windows Vista's Instant Search, etc.

2.1.2 Text Mining

Text mining is the process of extracting informative patterns or knowledge from unstructured text data. It is also known as text data mining or knowledge discovery from the textual corpus. Text mining is inevitable in microblog domain since most of the information available in microblog is text. However, the text data in microblog are usually informal noisy user-generated text i.e. inherently unstructured and fuzzy which makes the task complex. In general, text mining is a robust multidisciplinary field including information retrieval, information extraction, clustering, classification, visualization, text analysis, machine learning, and data mining [2].

A general framework for text mining is depicted in Figure 2.1. It has two main components: *Text refining* that transforms text documents into an *intermediate form* (IF) and *knowledge distillation* that infers knowledge or patterns from the *intermediate form* [2].

At first, *text refining* converts unstructured text documents into an intermediate form (IF). The intermediate form can be considered as either document-based or concept-based. *Knowledge distillation* from a document-based IF infers the knowledge or patterns across documents. By extracting object information relevant to a domain, a document-based IF can be projected onto a concept-based IF. *Knowledge distillation* from a concept-based IF infers knowledge or patterns across objects or concepts.

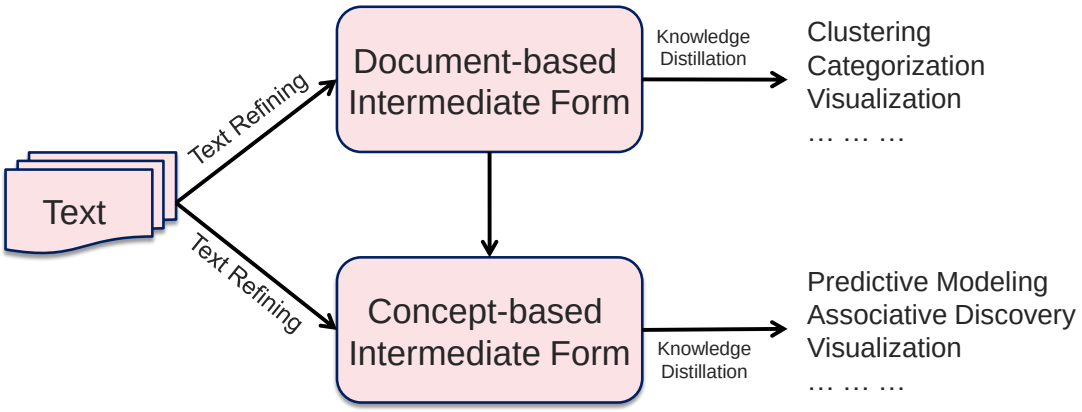

Figure 2.1: A general framework for text mining [1, 2].

For example, given a set of microblog documents, text refining first converts each document into a document-based IF. One can then perform knowledge distillation on the document-based IF for the purpose of organizing the document, according to their content, for visualization and navigation purposes. For knowledge discovery in a specific domain, the document-based IF of the microblog documents can be projected onto a concept-based IF depending on the task requirement.

2.1.3 Text Classification

With the rapid growth of microblog data available on the internet, it is inevitable for handling and organizing such amount of data. Text classification has become one of the key techniques for this purpose. Automated text classification approaches are applied to classify microblog documents, to find interesting information on the microblogging site, automatic indexing, content management, filtering [25], word sense disambiguation, and search space categorization [26, 27]. As it is difficult and time-consuming for building text classifiers by hand, it is advantageous to learn classifiers from sample data.

The major goal of text classification is to classify documents into a predefined set of categories. Each document may belong exactly one, or multiple, or no category at all. The objective of applying machine learning in this regard is to learn classifiers from sample data with labels, which is then used to assign the

class label of unknown documents. Assigning category by using this approach is called a supervised learning approach. Since categories may overlap, each category is treated as a separate binary classification problem.

The first step in text classification is to transform documents into tokens (strings or characters), that are suitable for the machine learning algorithm and the classification task. Information retrieval (IR) researchers recommend that word stems perform well as a representation unit and that their ordering in a document is of minor importance for many tasks. This issue leads to an attribute-value representation of text. Each distinct word w_i considered as a feature and the number of times word w_i available in the document is considered as a feature value. To avoid large feature dimension, words that are available in the training data at least 3 times are considered as features only if they are not "stop-words" (like and, or, etc.). This feature representation scheme reduces the very high-dimensional feature spaces that containing 10000 dimensions and more.

In brief, text classification in the field of information retrieval (IR) is an activity of labeling natural language texts with thematic categories from a predefined set. The standard methods of the machine learning techniques used in text classification usually operate on input documents after they have been transformed into feature vectors $f_1,\ f_2,\ ...,\ f_n\ \in\ D$. Most of the available techniques depend on the syntactic analysis of the features or keywords. They seldom analyze the semantics beneath the text.

2.1.4 Information Search in Microblog Domain

Microblog is an increasingly popular real-time information sharing platform among the people in the current world. Due to the length constraint of the microblogging platform, people usually express their thoughts in a concise manner in their own microblog account which can be read by his/her followers from anywhere in the world. Moreover, sometimes followers' broadcast the post to share the information in their own followers' domain. Due to the real-time nature of the microblogs, people usually search temporally relevant information and information related to people or entity [8, 28].

However, to satisfy their information need, people usually search in microblog using two ways [28]:

1. By expressing the information need as a query, conducting searches over pre-existing microblog data to find the relevant one.
2. Broadcasting the information need as a post to their followers so that they will answer them.

In the first approach, retrieving the relevant microblog based on existing microblog data is similar to traditional, ad-hoc information retrieval. But new approaches are required to tackle the microblog specific challenges such as real-time nature, temporally relevant information, huge vocabulary mismatch due to the informal noisy microblog texts, etc.

The real-time nature of the microblog search makes it more important and challenging in compared to the web search. For example, during a disaster period, the contents of microblog is treated as an important source to serve the situational information needs. Exploiting these situational information helps in reducing casualties, preventing secondary disasters, reduce economic losses, organize relief efforts, and social disruptions [29, 30].

2.2 Retrieval Models

Now, we briefly describe some classical information retrieval models used for traditional ad-hoc search including term frequency (TF), inverse document frequency (IDF), language model, Okapi BM25, divergence from randomness, and Jaro-Winkler similarity as follows:

2.2.1 Term Frequency (TF)

Term frequency (TF) means the frequency of a term in a document. The higher the TF, the higher the importance (weight) for the document [31].

Let, a tweet, T contains a set of word $\{w_1,\ w_2,\ ...,\ w_k\}$ with frequency $\{f_1,\ f_2,\ ...,\ f_k\}$, respectively. Then, term frequency is estimated as follows:

$$\text{Term frequency, } TF_i = \frac{f_i}{\sum_k f_k}$$

2.2.2 Inverse Document Frequency (IDF)

The inverse document frequency (IDF) is a measure of the general importance of the term that is obtained by dividing the total number of tweets by the number of tweets containing the term [31]. The IDF score of a term is estimated as follows:

$$\text{Inverse document frequency, } IDF_i = log\frac{|DT|}{|T : t_i \ \in \ T|}$$

where $|DT|$ is the total number of tweets in the corpus and $|T : t_i \ \in \ T|$ is the number of tweets where the term t_i appears. If the term is not appear in the corpus, this will lead to a division-by-zero. To overcome this limitation, it is therefore common to use $1 + |T : t_i \ \in \ T|$.

Sometimes, $TF - IDF$ is used to produce a composite weight for each term in each tweet.

$$TF - IDF = TF_i \times IDF$$

2.2.3 Vector Space Model

In the vector space model, tweets are represented as term vectors and the correlation between two vectors indicate the similarity between the two tweets [32]. If we express the query topic (Q) and tweet (T) as vectors:

$$\overrightarrow{T} = (w_1, w_2, w_3, ...w_n)$$
$$\overrightarrow{Q} = (w_1, w_2, w_3, ...w_n)$$

Then, the relevancy between a given query Q and tweet T can be estimated by cosine similarity (CosSim), which is defined as follows:

$$f_{CosSim}(\vec{Q}, \vec{T}) = \frac{\vec{Q} \cdot \vec{T}}{||\vec{Q}||||\vec{T}||} = \frac{\sum_{i=1}^{N} w_{i,Q} w_{i,T}}{\sqrt{\sum_{i=1}^{N} w_{i,Q}^2}\sqrt{\sum_{i=1}^{N} w_{i,T}^2}}$$

Cosine similarity is non-negative and bounded between [0,1], where 0 means 0% similar, and the 1 means 100% similar.

2.2.4 Language Model with Dirichlet Smoothing

In the language modeling approach, each document in the corpus is generated by the probability distribution over the terms in the vocabulary [33]. For a given query Q, retrieved tweet T is ranked by the likelihood of its corresponding language model.

$$f_{LM}(Q,T) = P(T|Q) \propto \ P(Q|T) \cdot P(T) \stackrel{\text{Rank}}{=} P(Q|T)$$

Assuming uniform priors over tweet documents and term independence:

$$P(Q|T) \ = \ \prod_{i=1}^{|Q|} P(w_i|T)$$

where $|Q|$ is the number of words in the query. Using multinomial language models, the maximum likelihood estimator of $P(w|T)$ will be:

$$P_{ml}(w|T) \ = \ \frac{n(w|T)}{|T|}$$

This estimate is then improved by using Dirichlet-smoothed language model as follows:

$$P(w|T) \ = \ \frac{|T|}{|T|+\mu} P_{ml}(w|T) + \frac{\mu}{|T|+\mu} P(w|C)$$

where $P(w|C)$ is the collections language model.

2.2.5 Okapi BM25 Model

Okapi BM25 model [34] is a bag-of-words retrieval function that measures the content relevancy between a query Q and a tweet T. The standard BM25 weighting function is formulated as follows:

$$f_{BM25}(Q,T) = \sum_{q_i \in Q} \frac{idf(q_i) \cdot tf(q_i,T) \cdot (k_1+1)}{tf(q_i,T) + k_1 \cdot (1 - b + b \cdot \frac{|T|}{avgtl})}$$

where $idf(q_i)$ is inverse document frequency, $tf(q_i,T)$ is the frequency of term q_i in tweet T, $|T|$ is the length of tweet T, and $avgtl$ stands for average length of tweet in the corpus.

2.2.6 Divergence from Randomness Model

Divergence from randomness (DFR) is a probabilistic approach which can be used as a query-dependent ranking model [35]. DFR models build upon the intuition that the more the content of a tweet diverges from a random distribution, the more informative the tweet is. Given a query Q and a tweet T, the standard DFR weighting function is formulated as:

$$f_{DFR}(Q,T) = \sum_{q_i \in Q} \frac{tf(q_i,T)\left(1 - \frac{tf(q_i,T)}{l_T}\right)^2}{tf(q_i,T)+1} \log_2 \left(\frac{tf(q_i,T)\bar{l}}{l_T tf(q_i,C)}\right) + 0.5 \log_2 \left(2\pi tf(q_i,T)\left(1 - \frac{tf(q_i,T)}{l_T}\right)\right)$$

where, $tf(q_i,T)$ is the occurrences of a term q_i in a tweet T, $tf(q_i,C)$ is the frequency of the term q_i in the corpus C, l_T is the length of tweet T, and $\bar{l}$ is the average length of all tweets in the corpus.

2.2.7 Jaro-Winkler Similarity

Given a query Q and a tweet T, the Jaro distance d_j of a query-tweet pair is estimated as follows [36]:

$$d_j = \begin{cases} 0, \text{ if m} = 0 \\ \frac{1}{3}\left(\frac{m}{|Q|} + \frac{m}{|T|} + \frac{m-t}{|m|}\right), \text{otherwise} \end{cases}$$

where m is the number of matching characters and t is the half of the number of transpositions.

However, Jaro-Winkler distance uses a prefix scale p which gives more favorable ratings to strings that match from the beginning for a set prefix length ℓ. Therefore, the Jaro-Winkler distance d_w between the query-tweet pair is estimated as follows:

$$d_w = d_j + (\ell p(1 - d_j))$$

where ℓ is the is the length of common prefix at the start of the string, p is a constant scaling factor used to adjust the score having common prefixes. The standard value for this constant in Winkler's work is $p = 0.1$.

2.3 Deep Learning Fundamentals

What is Deep Learning?

Deep learning is a sub-field of machine learning that trains the computer according to the nature of human. Therefore, deep learning algorithms are inspired by the structure and function of the human brain also called an artificial neural network (ANN). Like the human brain, ANN is consists of a collection of connected artificial neurons. These neurons are organized in layers. Each neuron within a layer is a mathematical function that takes the input data, transforms that data for further processing, and generate the output. A simple artificial neural network (ANN) is depicted in Figure 2.2.

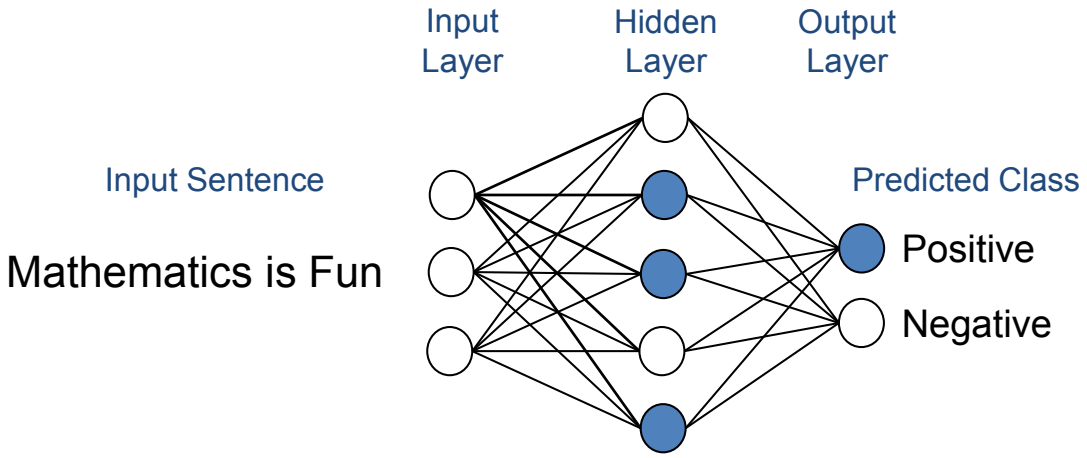

Figure 2.2: A simple artificial neural network (ANN).

This above simple diagram will provide a basic idea of how a simple neural network is structured. In this example, the network has been trained to identify the sentiment of the tweet, with the input layer being fed values, and the output layer predicting which sentiment category does the tweet belong. Each circle in the figure represents a neuron in the network. All neurons in the network are organized into vertical layers. Each neuron is linked to every neuron in the following layer, representing the fact that each neuron outputs a value into every neuron in the subsequent layer.

However, in a typical deep neural network (DNN), there will be more hidden layers of neurons between the input and output layers because the term “deep”

refers to the large number of hidden layers in the neural network. In short, models need to be trained with a large number of labeled data and the DNN architectures might contain many layers. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton presented a clear definition of deep learning with highlighting the multi-layered approach in their published paper titled as "Deep Learning" [37]:

"*Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.*"

Word Embedding

Nowadays, distributed representation of words known as word embedding is treated as one of the most popular representations of documents vocabulary due to its capability of capturing the context of a word within a document, estimating semantic and syntactic similarity as well as the relation with other words, etc. In general, word embedding is a mapping that associates words occurring in a collection to a vector space in R_n, where n is significantly lower than the size of the vocabulary of the collection [38]. Such vector space can help learning algorithms to achieve better performance in various natural language processing applications [39] such as query expansion [38, 40, 41, 42, 43], sentiment analysis [44, 45, 46], tweet ranking [47, 48, 49], and so on.

Previously different types of models including the well-known latent semantic analysis (LSA) and latent Dirichlet allocation (LDA) were used to estimate the continuous representations of words. However, recently distributed representations of words learned by neural networks gained popularity among the researchers because the learned vectors explicitly encode many linguistic regularities and patterns and each relationship is characterized by a relation-specific vector offset. Therefore, vector-oriented reasoning based on the offsets between words can be performed. For example, it was shown that vector("King") — vector("Man") + vector("Woman") results in a vector that is closest to the vector representation of the word "Queen" [50, 51].

To learn the distributed representations of words, Mikolov et al. [39] proposed two new model architectures including continuous bag-of-words (CBOW) and skip-gram model. The source codes are available in the word2vec software [52].

Continuous Bag-of-Words Model [3]: The continuous bag-of-words model known as CBOW tries to predict the current target word (i.e. the center word) based on the surrounding context words. According to the model architecture depicted in Figure 2.3, given the input $w(t-2), w(t-1), w(t+1), w(t+2)$ to a projection layer which is shared for all words, the output of the neural network will be $w(t)$.

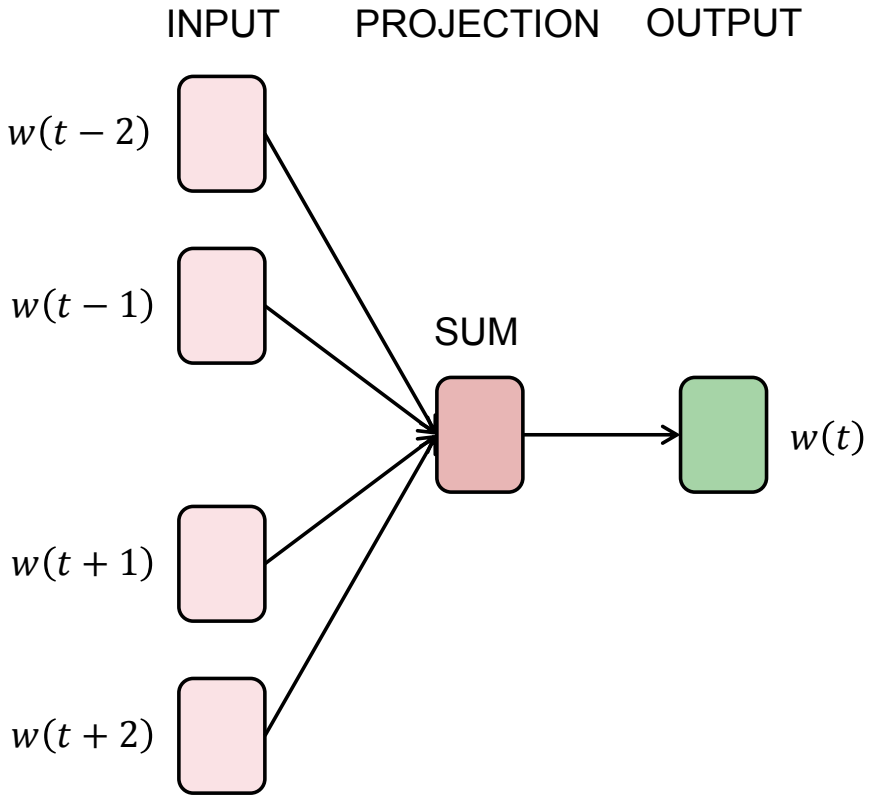

Figure 2.3: Continuous bag-of-words (CBOW) architecture proposed by Mikolov et al. [3].

Continuous skip-gram Model [3]: In contrast to the continuous bag-of-words (CBOW) model which predict the current word based on the context words, the continuous skip-gram model tries to maximize classification of a current word based on another word in the same sentence. Given each current word as an input to a log-linear classifier with continuous projection layer, this model will predict words within a certain range before and after the current word as depicted in Figure 2.4.

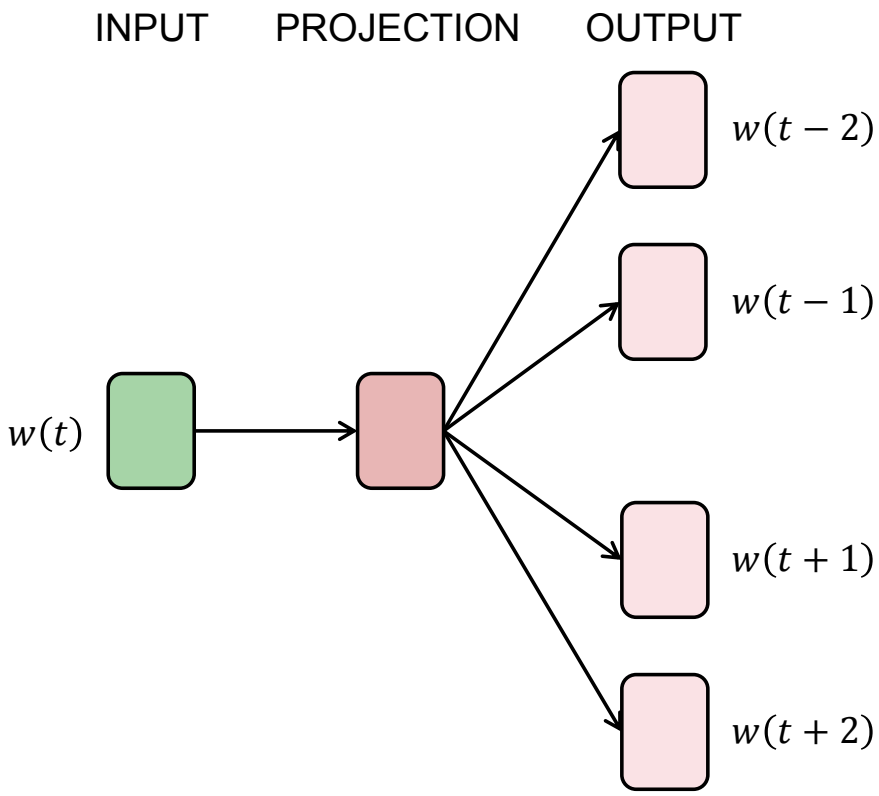

Figure 2.4: Skip-gram architecture proposed by Mikolov et al. [3].

In comparison, skip-gram works well with small amount of data and represent the rare words well, whereas CBOW is faster and has better representations for more frequent words [3].

Along with this direction, Stanford NLP team developed GloVe (Global Vectors) [53] for word representation where they used the word-word co-occurrence probability to build the embedding. The basic idea is that if two words are co-occurred many times, they may have similar meaning, therefore, the vectors generated by these words will be closer.

More recently, Mikolov et al. [54] improve the quality of the word vectors generated by word2vec model by combining the position dependent features proposed by Mnih and Kavukcuoglu [55], the phrase representation used in Mikolov et al. [39], and the subword information proposed by Bojanowski et al. [56]. This new architecture known as FastText word embedding model and increasingly gain popularity among the researchers.

Major Architectures of Deep Neural Networks

In this segment, we are going to describe two major neural network architectures including convolutional neural network (CNN) and long short-term memory (LSTM) network, that are used to develop several complex deep neural network (DNN) models.

Convolutional Neural Network (CNN)

Convolutional neural networks (CNN) has recently achieved remarkable performance improvement in various natural language processing (NLP) applications especially in sentence modeling and classification [57, 58, 59]. CNN use the layers with convolving filters that are applied to local features [57]. Next, we will describe a basic CNN model that can be applied for sentiment classification of a sentence.

As depicted in Figure 2.5, at first, a tokenized sentence is converted to a sentence matrix by using a pre-trained word2vec or glove model. The rows of the matrix are the word vector representations generated from each token. If we consider the length of a sentence is L and word-vector dimension is D, then the dimensionality of the sentence matrix will be $L \times D$. According to the Collobert and Weston [60], we then considered the sentence matrix as an image and perform the convolution on it by using a filter. One can use different kinds of filters with the different height of the filter i.e. window size. Each filter generated the corresponding feature map which dimensionality may vary according to the filter type and window size. A max pooling function is then applied to extract the scalar from each feature map and concatenate them to form a fixed length top-level feature vector. This feature vector then fed into a softmax layer to generate the final prediction. Following the recommendation by Hinton et al. [61], one may apply dropout as a means of regularization at the softmax layer. For the training purpose, one may usually consider the categorical-cross-entropy as the loss function and train the model by minimizing the error. Optimization is performed using SGD and back-propagation [62].

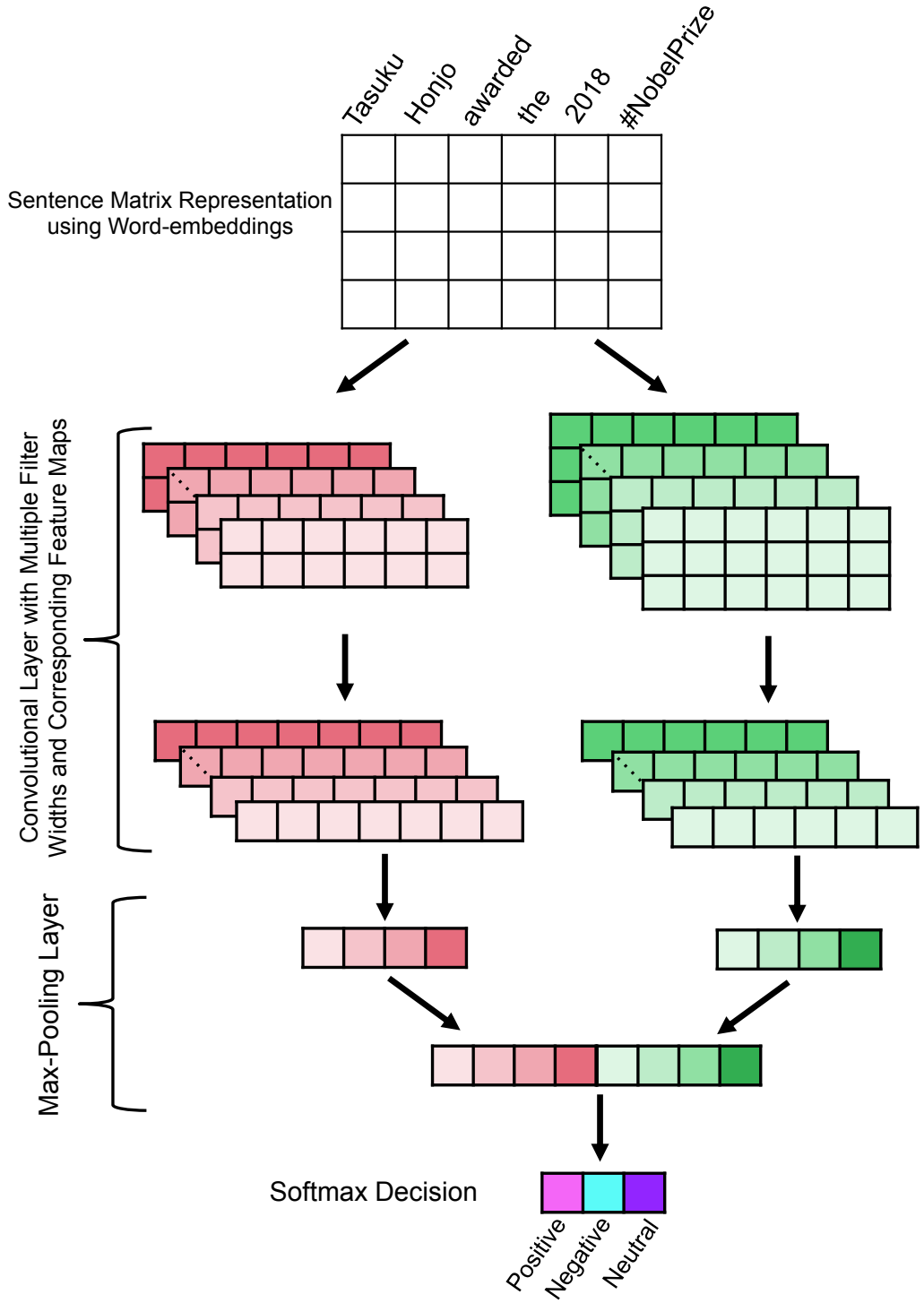

Figure 2.5: Illustration of a convolutional neural network (CNN) architecture.

Long Short Term Memory (LSTM) Network

Recurrent neural networks (RNN) were developed to learn long-term dependencies and help us to deal with the sequences of variable length. However, in practice, RNNs are limited to learn short-term dependencies due to the vanishing gradient problem. To overcome this limitation Hochreiter and Schmidhuber [63] were first introduced the long short-term memory (LSTM).

Given an input sequence $(x_1; x_2; ....x_N)$, LSTM computes the hidden vector sequence $(h_1; h_2; ....h_N)$ and the output vector sequence $(y_1; y_2; ....y_N)$. The building block of an LSTM is a memory cell. In an LSTM cell, there are three different types of gates including input gate, forget gate, and output gate. These gates are collectively decide the transitions of the current memory cell c_t and the current hidden state h_t [64, 65]. The LSTM transition functions are defined as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
u_t &= \phi(W_u \cdot [h_{t-1}, x_t] + b_u) \\
c_t &= f_t \odot c_{t-1} + i_t \odot u_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where i_t, f_t, o_t, u_t, c_t, and h_t denotes the input gate, forget gate, output gate, cell input activation, the cell state, and the current hidden state, respectively, at the current time step t. The symbol σ is the logistic sigmoid function to set the gating values in $[0, 1]$. ϕ is the hyperbolic tangent activation function that has an output in $[1, -1]$ and $\odot$ is the element-wise multiplication.

At the last time step of LSTM, the output of the hidden state is regarded as the tweet representation and passed to a fully connected softmax layer on top. Following the recommendation by Hinton et al. [61], one may apply dropout as a means of regularization at the softmax layer. For the training purpose, one may usually consider the categorical-cross-entropy as the loss function and train the model by minimizing the error. Optimization is performed using SGD and back-propagation [62]. A simple illustration is depicted in Figure 2.6.

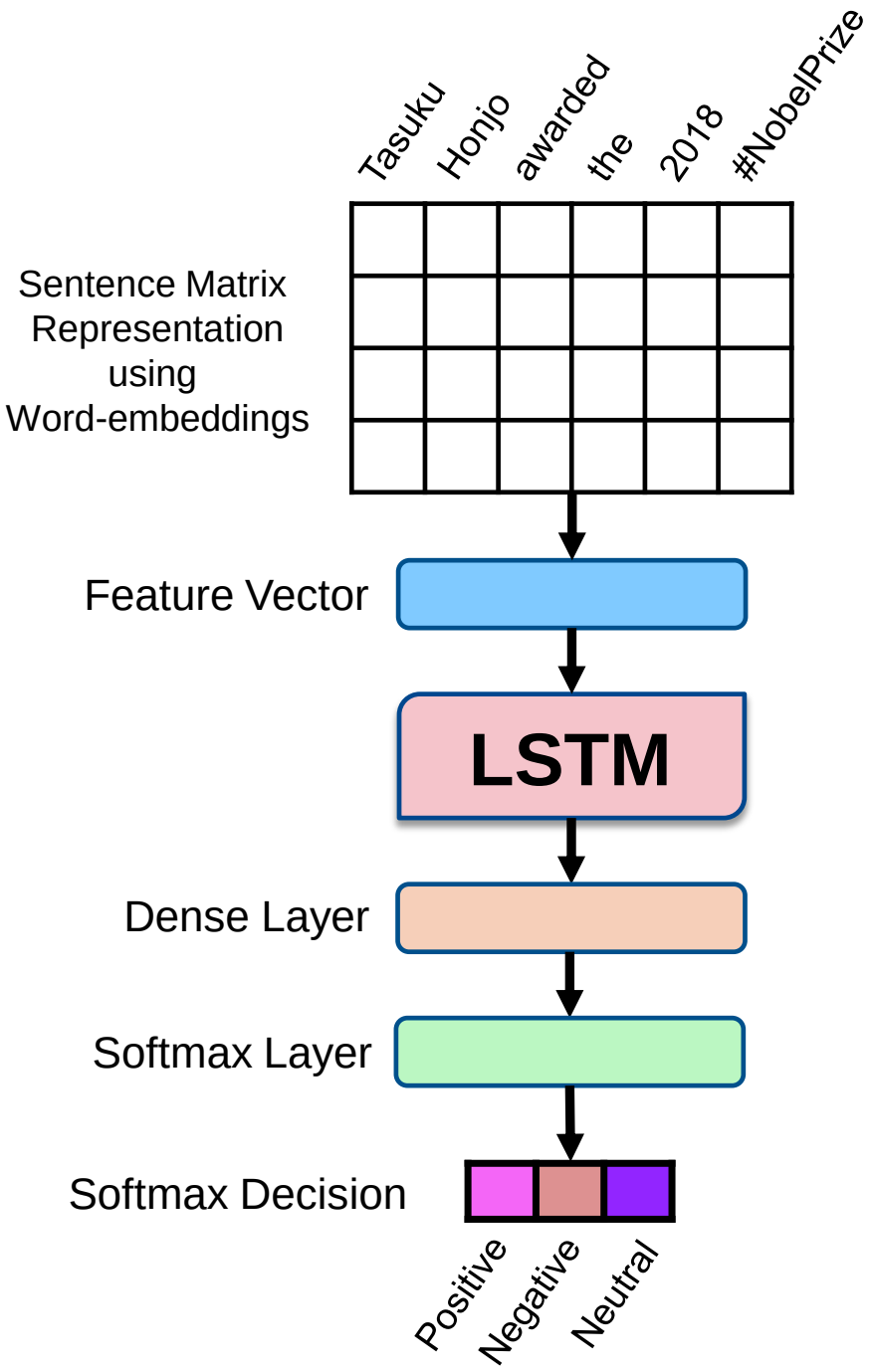

Figure 2.6: Illustration of a long short-term memory (LSTM) network architecture.

Deep Learning Applications

Recently, deep learning models achieved state-of-the-art performance and sometimes exceeded the human-level performance in various real-world applications. Famous deep learning specialist Andrew NG said that [66]:

"Deep Learning is a superpower. With it you can make a computer see, synthesize novel art, translate languages, render a medical diagnosis, or build pieces of a car that can drive itself. If that isn't a superpower, I don't know what is."

In this segment, we will discuss some applications of deep learning techniques on several ongoing information retrieval (IR) and natural language processing (NLP) researches and fields.

The web is rapidly moving towards a platform for mass collaboration in content production and consumption, and the increasing number of people are turning to these online sources for sharing their opinions and want to satisfy their information need. Therefore, individuals and organizations are increasingly seeking ways to analyze these huge data due to its wide range of applications including market research, business intelligence, enhancement of online shopping infrastructures, predicting the stock market, political polls, and so on [67].

In this regard, several types of research are conducted by the researchers. One of them is the retrieval of the relevant documents based on a user query, where deep learning techniques are successfully applied to improve the performances. Some researchers utilized the deep neural network (DNN) based models to improve the performance of the ranking models [68, 69, 70, 71, 72], where some others utilized it for selecting candidate expansion terms for query expansion [38, 40, 41, 42]. Moreover, deep learning also successfully applied to diversify the search results as well as vertical based search [73, 74, 75]. More recently, various web platforms especially microblog is treated as an important source to serve the situational information needs during a disaster period. In this regard, deep learning technologies are successfully applied for monitoring and producing the curated contents based on different information types from massive microblog posts which provide enormous opportunities to different public safety personnel or used for post-incident analysis [30, 76, 77, 78].

Along with this direction, deep learning also achieved the significant improvement in the field of opinion mining, which is the computational study of people's opinions, sentiments, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. Currently almost every state-of-the-art models utilized the deep learning technologies to achieve the improved performance in various opinion mining tasks including, sentiment analysis [44, 65, 79, 80], stance detection [81, 82, 83, 84], emotion analysis [85, 86], emotional state estimation [87, 88], emoji prediction [89, 90], subjectivity analysis [57], etc.

Another successful application of deep learning is anomaly detection. For example, retailers frequently have to deal with some thief customers who use stolen credit cards to make excessive orders or customers that retract payments procedures once products have already been delivered. Currently, deep learning techniques achieved amazing improvement in the anomaly detection task [91, 92].

Recent online services exploiting deep learning techniques heavily to deal with the automatic personalization of a large number of users to recommend the relevant contents. Researchers try to utilize the users' behavioral aspects, web browsing history and search queries, as well as many other related features in their deep learning framework to design a robust recommendation system [93, 94, 95].

Moreover, in the language identification task, deep learning has achieved dramatic improvements over the other traditional models [96, 97]. Another successful application of deep learning is in the field of automatic machine translation, where given words, phrase or sentence in one language, automatically translate it into another language [98, 99, 100].

Automatic text generation is another interesting task, where a corpus of text is learned to generate the new text, word-by-word or character-by-character. To achieve this goal, large DNN models are used to learn the relationship between the sequences of input strings and generate the text accordingly [101, 102, 103]. The trained model can eventually learned how to spell, punctuate, form sentences, and even capture the style of the text from the training corpus. The automatic text generation techniques can be applied to several domains including poem generation task, sentence completion task, author identification task, and so on.

Chapter 3

Literature Review

3.1 Blog Search

Many approaches have already been developed for blog search and forum search, which basically include learning to rank based methods and link-based method [11]. A brief articulation of the relevant study is presented next.

3.1.1 Learning to Rank Approach

Xi et al. [104] proposed an approach to predict the most relevant messages in community search based on a user' query. In this regard, they utilize the features from the thread trees of newsgroup messages, authors and lexical distribution within a message thread and trained the linear regression (LR) and support vector machine (SVM) for the ranking. Whereas, Fujimura et al. [105] proposed an "EigenRumor" algorithm for ranking blogs that exploited the provisioning link and evaluation link between bloggers and blog entries, and ranked the blog entry by weighting the hub and authority scores of the bloggers. In another approach, Han et al. [106] present a blog ranking framework, named PTRank, that utilized the relevance feedback from users as well as various information that are available from RSS feeds to improve the search quality. To estimate the relevance score between a keyword and a blog post, a neural network method is employed to learn a relevance scoring functions.

3.1.2 Link based Approach

Kritikopoulos et al. [107] proposed a method to rank weblogs based on the link graph and on several similarity characteristics between weblogs. To estimate the similarities among bloggers and blogs they assigned the importance score to the blog entry based on the bloggers' popularity. On the other hand, Xu and Ma [108] proposed Fine-grained Rank (FGRank) method that utilized a topic hierarchy structure built through content similarity. In another approach, Liu et al. [109] introduced a structure-based ranking approach, named PostRank, which utilized the posting trees that built according to response relationship between postings. Along with this direction, Chen et al. [110] proposed a posting rank algorithm that exploited the common responders between postings and leveraged the relationship between these common repliers through building implicit links based on that co-repliers relation and construct a link graph. To search in the business blog domain, Chen et al. [111] proposed a probabilistic models by exploiting latent semantic analysis (LSA) and probabilistic latent semantic analysis (PLSA) to analyze the problems of synonymy and polysemy in the blog search. Joshi et al. [112] demonstrated a blog mining and search framework, named BlogHarvest, that extracted the interests of the blogger, finds and recommends blogs with similar topics and provides blog oriented search functionality. In this regard, they used the classification techniques, linkage and topic similarity based clustering as well as POS tagging based opinion mining. Kuwata et al. [113] proposed a method to find the right product reviews for consumers from blog search. They first extracted whether documents of blog site include review sentence or not. The method comprised of two steps. First, it needs to set up a feature for each product. It then sends different features to the blog search engine and collects all blog posts to produce more product reviews in the second step.

However, approaches discussed above are not effective for microblog search (i.e. twitter search) because microblog posts are usually short in length and contain informal user-generated contents which exacerbate severe vocabulary mismatch problems compared to the blog posts. Moreover, the real-time nature of the microblogs and other microblog specific features contribute significantly to the microblog retrieval.

3.2 Microblog Search

Nowadays microblog search is one of the hot research topics in the field of information retrieval (IR). Among several microblogging platforms, most researches currently focus on Twitter. Microblogs have specific characteristics that introduce new problems and challenges for retrieval task [8, 28].

Although twitter maintains a specialized search engine which ranks tweets according to posting time and topic popularity, a number of web platforms provide the real-time microblog search service based on a users query. These systems utilized the posting time, account authority, topic popularity, and content relevance or similarity score to provide the ranked search results [11]. Names of some of these systems are Twazzup[1], Chirrps[2], Tweefind[3], Twitority[4], CrowdEye[5], etc.

However, these systems didn't address all the challenges of the microblog retrieval and performance of these so-called systems didn't satisfy the user information need properly. Considering this Ounis et al. [6] introduced the TREC microblog ad-hoc search task in 2011, where a user's information need had been represented by a query at a specific point in time and a set of relevant tweets had been returned. This ad-hoc task continues until 2015, wherein 2011-2012 [6, 114] the organizers used a small microblog corpus consisted of 16 million tweets and 2013-2014 [115, 116] the organizers used a large corpus consisted of 253 million tweets.

After that, several methods are exploited by many researchers from 2011 to 2018 to address the challenges of microblog retrieval. For the simplicity of discussion, we can broadly categorize these studies into two groups: (1) improve the performance of microblog search by reranking the initial retrieved tweets; (2) utilize the query expansion technique to augment original query representation so that the augmented query can retrieve more relevant tweets.

In both types of approaches, researchers were leveraged the temporal aspect of twitter posts (i.e. recency and bursty nature), twitter specific characteristics

[1]http://www.twazzup.com/
[2]http://chirrps.com/
[3]http://www.tweefind.com/
[4]http://www.twitority.com/
[5]http://www.crowdeye.com/

(such as hashtags, retweets, existence of hyperlinks, etc.), external (e.g. web page, Wikipedia, freebase) resources to estimate the relevance, modern and representative information retrieval models, quality indicators for tweet text, user behavior model, and so on. Next, we will discuss some prominent works for each of the approaches.

3.2.1 Reranking based Approaches

Information retrieval in microblog environment, such as twitter, is one of the state of the art research task, where the major goal is to return a ranked list of tweet documents based on the user's query. Prior research on microblog post retrieval indicates that retweet removal, future tweet removal, spam removal, and unwanted languages tweet removal improved the retrieval performance significantly [117].

Kanhabua et al. [118] classified existing time-aware ranking approaches into recency based ranking and time-dependent ranking. Recency based ranking has treated the recently created documents as relevant. People usually search microblog posts for real-time information need [8], therefore recency is considered as an important temporal property for retrieving relevant tweets [13, 119, 120, 121]. On the other hand, time-dependent ranking approach considers the relevant time periods underlying a query. Following this direction, Jia et al. [14] estimate the temporal relevance score according to the categorization of queries based on the temporal distributions of their top-retrieved tweets.

Modern and representative retrieval models, including inverse document frequency (IDF), Okapi BM25, language model, vector space model, probability ranking principle (PRP), etc. also utilized by several researchers [11, 122, 123] to estimate the content relevancy. Nowadays word embedding based features are also used for enhancing retrieval effectiveness [48, 124]. Like content aware and temporal features, other twitter related features, including URL count, retweet count, and hashtag score are also proposed by several researchers [11, 125]. Severyn et al. [126] reported that relational syntactic features generated by structural kernels are effective for learning to rank (L2R) algorithm.

Since tweets are limited in length and the average length of the queries in microblog is about 1.64 words, the vocabulary mismatch problem exacerbates the difficulty of query term matching during the retrieval [127]. Some researchers [13, 120, 122, 128] address the vocabulary mismatch problem by expanding the representation of the queries. Their results indicate that significant improvements in retrieval effectiveness can be achieved by employing query expansion (QE) methods. Along with this direction, some researchers address this problem by incorporating document expansion with QE [127].

Recently, Choi et al. [129] have proposed a user behavior based quality model to indicate the correlation between tweet document informativeness and relevance judgments. Fan et al. [130] proposed a feedback entity model and integrated it into an adaptive language modeling framework in order to improve the retrieval performance. Rodriguez et al. [131] considered a microblog document as a high-dimensional entity and reported that the relative presence of the different dimensions within a document and their ordering are connected with the relevance of microblogs.

To select the best set of features, supervised feature selection approaches based on learning to rank algorithm [11], LASSO, and Elastic-net regularization method [132] are employed by several researchers. There are several previous works employing feature based machine learning approach for tweet ranking [12, 11]. In order to address the real-time tweet retrieval problem, Metzler et al. [12] made use of feature based RankSVM to rerank the retrieved tweets with respect to queries. This work achieved the best results reported in TREC 2011.

3.2.2 Query Expansion based Approaches

In microblog search, given a users' query, a set of relevant tweets are provided to satisfy the users' real-time information need. However, the tweets being short in length often contains unconventional word forms and queries provided by users are usually too short, ambiguous, and hardly describe the information need accurately, which may lead to unsatisfactory results [15]. A widely used solution to this problem is the query expansion (QE), which augments the original queries with terms that best represent the users' intent.

Among several query expansion methods, pseudo-relevance feedback (PRF) based approaches were widely studied in microblog retrieval [15, 16, 17, 47]. The PRF approach assumes that top-ranked documents of the original query based on initial retrieval results are relevant and contain terms related to the query intent to augment the query representation. Albishre et al. [15] proposed a PRF model which considered discriminative expansion to meet the user interests. El-Ganainy et al. [16] proposed a hyperlink-extended PRF that utilized the presence of embedded hyperlinks in retrieved microblogs. Zingla et al. [17] proposed a technique that extracted semantically related expansion terms from Wikipedia, DBpedia, and unstructured texts. Chy et al. [47] used a PRF based simple query expansion by leveraging external resources and focused mainly on reranking the initial retrieved tweets based on several features by estimating the relevance of query-tweet pair.

People usually search microblog posts for real-time information need [8], therefore incorporating temporal property of terms with the query expansion approach improving the performance of microblog retrieval [18, 19, 119]. Miyanishi et al. [19] proposed a time-based query expansion (QE) method that can handle the recency and temporal variation according to the topic's temporal variation. In another work [122], they proposed a two-stage PRF model using manual tweet selection to improve the initial retrieval results and integrated the lexical and temporal evidence into the model. Massoudi et al. [120] proposed a dynamic query expansion model, where they showed that temporally closer terms in response to query time are more effective for query expansion. Rao and Lin [18] utilized the continuous hidden Markov model (cHMM) to identify the bursty temporal clusters where tweets in the bursty states were selected for query expansion. Wang et al. [133] utilized both lexical and temporal expansions to improve the performance of the query expansion model.

Topic modeling is employed by some researchers to uncover hidden topics within tweets and utilized it for query expansion [15, 120, 134]. However, conventional topic models such as latent Dirichlet allocation (LDA) [135] and probabilistic latent semantic analysis (PLSA) [136, 137] suffer from the severe data sparsity problem due to the lack of word frequency and contextual information in tweets. To alleviate this problem, Yan et al. [138] proposed a biterm topic model

(BTM) where the topics are learned over tweets by directly modeling the generation of biterms in the given corpus. Besides, microblog users tend to use entities in their queries to express their information need. Considering this fact, Fan et al. [130] leveraged the rich entity information in twitter in their proposed feedback entity model and incorporated it into an adaptive language modeling framework to enhance the performance of microblog search. In this direction, #hashtag is considered as a user-generated entity and used for query expansion [139].

More recently, term co-occurrence based embeddings such as *word2vec* [39] and *GloVe* [53] were investigated to enhance the performance of the IR system [48, 124, 140]. Since the objective of query expansion is to expand the query with semantically relevant terms, some researchers leveraged word embeddings to improve the QE performance in Web search [38, 40, 41, 42, 43].

Along with this direction, deep learning models such as convolutional neural network (CNN) [57, 79] and long short-term memory (LSTM) [63, 141] model have achieved significant improvements in document modeling. Though CNN is able to learn local response from temporal or spatial data, it has the limitation of learning sequential correlations. To overcome this limitation, some researchers used the combination of CNN and LSTM called C-LSTM [65, 142] to capture the benefits of both architectures.

In Web search, learning to rank (L2R) methods had been used by several researchers [20, 21, 143, 144] to rank the candidate expansion terms. In addition, supervised feature selection approaches based on learning to rank algorithm [11, 21] and elastic-net regularization method [47] were employed by researchers to select the effective set of features.

In summary, we identify several limitations of the existing works for both the reranking and query expansion based approaches. In the reranking based approaches, we realize the absence of a method that effectively exploits the temporal property and other twitter specific characteristics in a unified framework. Along with this direction, in the query expansion based approaches, most of the methods utilized only the PRF tweets to select the expansion terms and didn't utilize the rich set of query-term relevance features. In our proposed approach described in Chapter 4 and Chapter 5, we have tackled most of these limitations to improve the performance of microblog search.

3.3 Microbog Retrieval during Emergencies

Microblog, especially twitter, is treated as an important source to serve the situational information needs during a disaster period. Monitoring and producing the curated tweets based on different information types from massive twitter posts provide enormous opportunities to different public safety personnel or used for post-incident analysis [145].

To effectively utilize microblogging sites during disaster events, Rudra et al. [146] proposed a framework that classifies the tweets to extract situational information and summarizes the information. Their framework takes into consideration of the typicalities pertaining to disaster events whether a tweet contains a mixture of situational and non-situational information along with certain numerical information. Truong et al. [29] proposed a Bayesian approach to identify the informational tweets from the tweet streams during a disaster period. Dutt et al. [147] proposed a system named as SAVITR for extracting real-time location information from microblogs during emergencies.

Moreover, Gosh et al. [76] introduced a task at the 2016 forum of information retrieval (FIRE), which goal is to address the challenges of retrieving specific types of situational information from the twitter posts during the disaster period. Basu et al. [30, 77] conducted a comparative performance evaluation of the participants' systems as well as traditional IR models for this task. Along with this direction, in the 2017 FIRE microblog track, Basu et al [78] introduced a task to identify only the need tweets and availability tweets from the tweet streams. The need and availability tweets are very important for coordinating relief operations in a disaster situation. Many teams have participated in these tasks with their proposed solutions to tackle the challenges.

More recently, at the 2018 text retrieval conference (TREC), McCreadie et al. [145] introduced an incident stream (TREC-IS) task which is designed to tackle the challenges of microblog retrieval during a disaster period. The main task for the 2018 TREC-IS track was to categorize the tweets in each event/incident's stream into different high-level information types that are defined in the TREC-IS incident ontology. The data were sampled from a variety of incidents such as earthquakes, hurricanes, shootings, typhoon, etc.

3.4 Microblog Recommendation

With the emerging popularity of the microblog platform, researchers of the recommendation system community are increasingly interested to employ the recommendation techniques such as collaborative filtering to predict the users' tweets preference and provide the users' most relevant tweets according to his interest.

Chen et al. [148] proposed a collaborative ranking based approach that capturing personal interests for tweet recommendations. They considered the tweet topic level factors, user-social relation factors, and explicit features such as authority of the publisher and quality of the tweet. Diaz-Aviles et al. [149] also used the collaborative filtering in their proposed approach that recommended hashtags on twitter in real-time. Whereas Bedi et al. [150] proposed an extreme learning machine based recommendation (ELMR) technique by exploiting user behavior patterns and interactions (based on retweets and favorites) to provide the online tweet recommendation for the movies. Yu et al. [151] extended the session-based temporal graph (STG) approach which utilized the textual information, the time factor, and the users' behavior features in twitter for tweet recommendation. As users are increasingly concerned about their privacy, Liu et al. [152] proposed a privacy-preserving personalized tweet recommendation framework, named as PTwitterRec, that provided the personalized tweet recommendations while keeping users' tweets and interests hidden from the online social networks (OSN) provider as well as other unauthorized entities.

More recently, Harakawa et al. [153] utilized the multimodal field-aware factorization machines (FFM) to design a sentiment-aware personalized tweet recommendation. Karidi et al. [154] proposed a semantic tweet recommendation method that utilized the knowledge graph (KG) which represented all user topics of interest and the relations between them. Zeng et al. [155] proposed a microblog conversation recommendation system by introducing a unified statistical learning framework that jointly learned the hidden factors which reflected the user interests. To perform the mention recommendation in twitter, Huang et al. [156] incorporated the interests of users and utilized the cross-attention mechanism to extract both textual and visual information in their proposed cross-attention memory network.

Since tweets contain a wide variety of information about an important event, a huge number of twitter posts might contain irrelevant and redundant information [157]. To overcome this limitation, automatic summarization may play an important role to design an effective summarization system. By employing the summarization technique, it is possible to select the few messages that cover all the information related to the event according to the users' interest.

Ren et al. [158] proposed a time-aware tweets summarization system by exploiting the users' history and collaborative social influences from "social circles". In addition, they introduced a time-aware user behavior model, named as tweet propagation model (TPM) that infer dynamic probabilistic distributions over interests and topics. Li et al. [159] utilized the social media's check-in histories and complaint discovery about water management in social media in their proposed geo-spatial profile summarization system. Zubiaga et al. [160] proposed a two-step process for the real-time summarization of events: (1) sub-event detection and (2) tweet selection without making use of external knowledge. Shou et al. [161] proposed a tweet summarization prototype called Sumblr (SUMmarization By stream cLusteRing), which consisted of two main components, namely a tweet stream clustering module and a high-level summarization module. Xu et al. [157] proposed an event-graph based method based on information extraction techniques to create tweet summaries of variable length for different topics by extending a PageRank-like algorithm to partition event graphs. Rakesh et al. [162] proposed a framework to identify and summarize tweets that are specific to a location.

Moreover, considering the importance of tweet summarization, Lin et al. [116] introduced the tweet timeline generation (TTG) task at the TREC-2014, where the goal is to produce concise summaries of the posts that are relevant to the given query. Along with this direction, they introduced the real-time summarization (RTS) task [163] at TREC-2016 that was intended to explore techniques and systems to monitor streams of twitter posts and keep users up to date by providing (i.e. recommend or suggest) interesting and novel contents in a timely fashion. Several participants joined in these tasks and explore several techniques to improve the summarization performance from informal twitter posts.

Chapter 4

Time and Context Aware Microblog Reranking Approach

4.1 Introduction

Nowadays, microblog web sites are not only the places in maintaining the social relationships, but also act as a valuable information source. Everyday lots of users turn into microblog sites for sharing their views, opinions, experiences, important news, and also want to get some information what is happening around the world. Among several microblog sites, Twitter[1] is now the most popular, where lots of users post tweets whenever a notable event occurs. Hence, information retrieval in twitter has made a hit with a lot of complaisance. By searching tweets, users find temporally relevant information, such as breaking news and real-time events [8]. That means, freshness (i.e. recency) of the tweet with respect to query time is an important factor of relevance. Another important characteristic of twitter is that people tends to post about a topic within a specific period of time (i.e. bursty nature). For example, when the breakup news of famous band "White Stripes" published on 2nd Feb, 2011, many people post tweets about this topic on that day. That is why; posts that are generated before or after this date are less relevant to the query, "White Stripes breakup." Morevoer, due to the length constraint of tweets, peoples usually use unconventional abbreviations, poor linguistic phrases,

[1]https://twitter.com

and URL in their tweets. Hence, the vocabulary mismatch problem between a query-tweet pair becomes worrisome for effective IR over tweets. Besides, twitter uses the specific syntax (e.g. re-tweets, hashtags), that also poses the challenge to conventional IR techniques. For addressing such kinds of twitter characteristics and boosting the retrieval performance, TREC was introduced the microblog ad-hoc search task in 2011 [6], where a user's information need had been represented by a query at a specific point in time and a set of relevant ranked tweets had been returned.

In this chapter, we have proposed a method to rerank the tweets that are retrieved using a baseline method. To achieve this, we consider content relevance features, twitter specific features, account related features, context relevance features, popularity based features, and temporal features. Moreover, automatic query expansion, supervised feature selection, and ensemble of machine learning techniques are also applied. Experimental results with TREC microblog dataset showed that our method improves the retrieval performance over the baseline and known related methods [12, 13, 18, 19, 164, 165].

The main contributions of our proposed approach are as follows: 1) To address the temporal aspects (recency and temporal variations) of tweets, we introduce two effective temporal features. 2) To overcome the limitations of the vocabulary mismatch problem, we introduce four context relevance features based on word-embedding and query-tweet sentiment correlation. In this context, we also introduce a simple but effective three-stage query expansion technique. 3) To determine the queries temporal and sentiment sensitivity, we introduce a query type determination technique. 4) To estimate the importance of tweets, we introduce our own version of URL popularity and hashtag importance features.

The rest of the chapter is structured as follows: **Section 4.2**, describes in detail of our proposed tweet reranker system. **Section 4.3** includes experiments and evaluation as well as the comparisons with the state-of-arts to show the effectiveness of our proposed method. To conclude the chapter, we present a summary of our proposed approach and some tentative future directions to overcome the limitations of our approach in **Section 4.4**.

4.2 Proposed Microblog Reranker Framework

Now, we describe the details of our proposed method. The goal of our tweet reranker system is to rank the tweets that are retrieved by using a baseline method. The overview of our proposed framework is depicted in Figure 4.1.

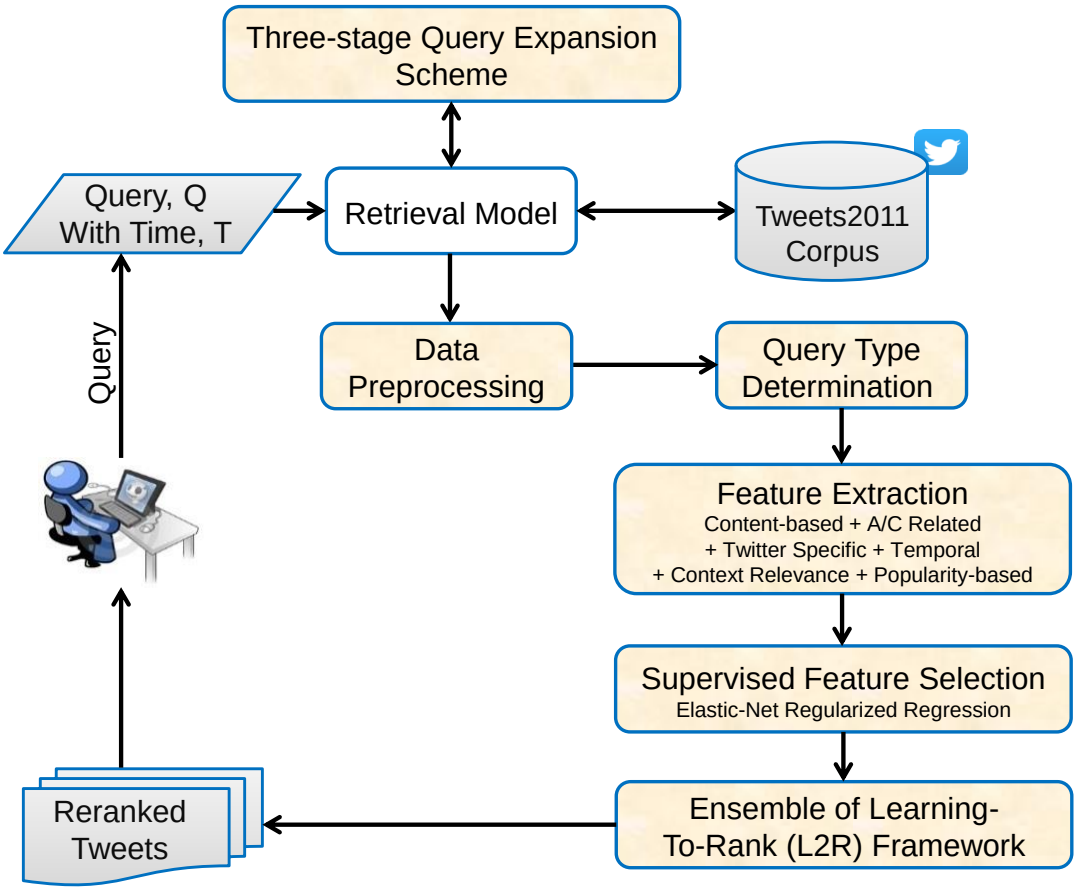

Figure 4.1: Proposed microblog reranker framework.

We first fetch 1000 tweets for each query topic by using the baseline method. A three-stage query expansion scheme formulates the query to fetch 1000 tweets again. In the preprocessing stage, we perform lexical normalization, non-English tweets removal, retweets removal, and future tweets removal. To extract temporal and sentiment aspect of tweets, we determine the temporal and the sentiment dimension of queries by utilizing the temporal and sentiment distribution of top

retrieved tweets. In the feature extraction stage, we extract several effective features broadly grouped into six different categories, including content relevance features, twitter specific features, account related features, context relevance features, popularity based features, and temporal features. To scale the feature value, we make use of the *MinMax* [166] normalization technique. We also apply a supervised feature selection method based on *elastic-net* regularization to select the best features combination. In order to estimate the importance of the selected features, we apply random forest as a feature ranking method. To estimate the relevancy of query-tweet pair, we make use of the ensemble of learning to rank (L2R) framework.

4.2.1 Data Preprocessing

In the preprocessing stage, a filtering process has been applied to refine the crawled results based on retweet removal, non-English tweet removal, and future tweet removal. Tweets that begin with the word of "RT" are regarded as retweets and eliminated from the corpus with the consideration that they are just the identical copy of other tweets without any useful information. Though twitter is a multilingual microblog environment, in this research non-English tweets are judged non-relevant. To remove the non-English tweet from the corpus, we apply a language detection library[1]. In additions, tweets often contain unconventional word forms and domain-specific entities. For example: "2day" instead of "today", "Birrrtthhdaayy" instead of "Birthday", "Congratz" instead of "Congrats", etc. To normalize such kind of non-standard words into their canonical forms, we utilize two lexical normalization dictionaries collected from [167] and [168].

Moreover, we also remove the non-English characters from tweets. Tweets that are posted after the query timestamp are treated as future tweets and removed from the corpus. As tweets are very short in length, we do not remove stop-word from tweet text during our experiment except query expansion. For stopword removal, we applied the Indri's standard stoplist[2].

[1]https://code.google.com/p/language-detection/

[2]http://www.lemurproject.org/stopwords/stoplist.dft

4.2.2 Query Expansion

The objective of our three-stage query expansion approach is to alleviate the vocabulary mismatch problem. It is the process of reformulating original query by enriching it with additional words. To expand the query, we utilize the pseudo relevance feedback (PRF) approach in the top retrieved tweets at the first stage, make use of web search results at the second stage, and extracting relevant hashtags from the top retrieved tweets at the third stage of query expansion as depicted in Figure 4.2.

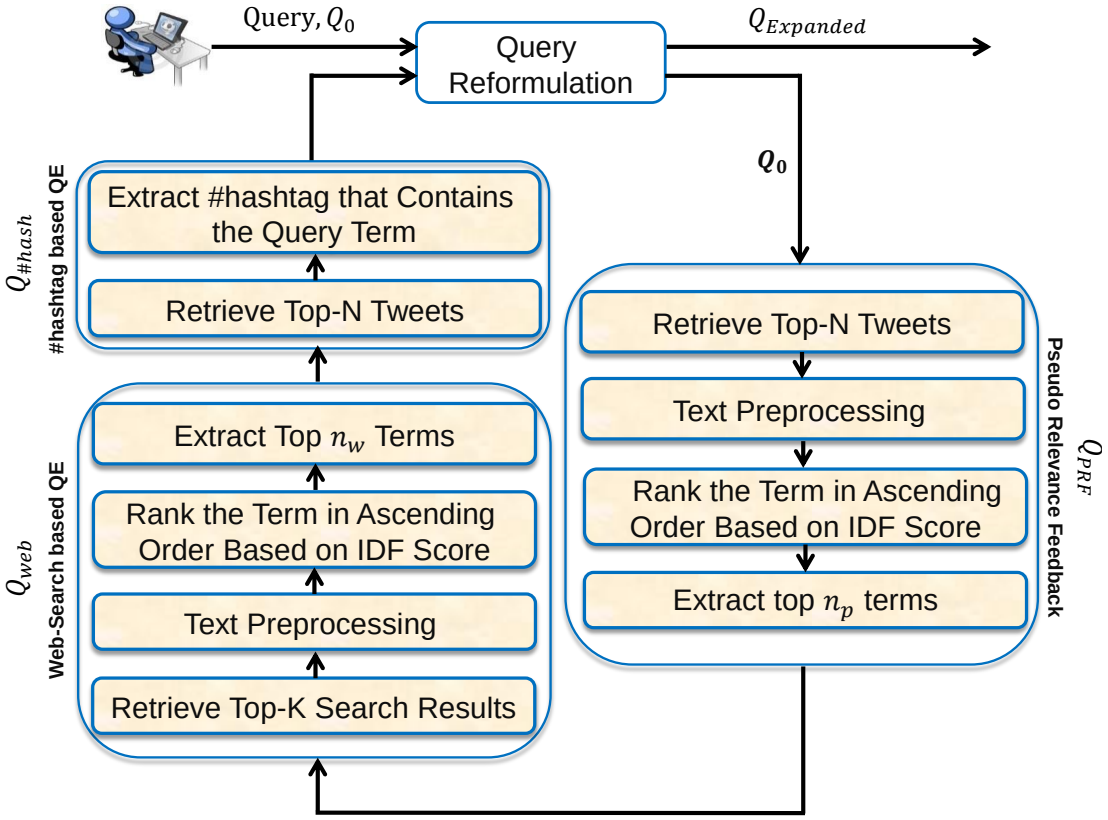

Figure 4.2: Three-stage query expansion framework.

In PRF approach, the top-N retrieved tweets in response to the original query Q_0, are treated as relevant, because terms within these tweets have greater probabilities to retrieve relevant tweets within that particular topic. To select the top n_p terms for expansion that are not in the original query, IDF-score of each term is considered and referred to as Q_{PRF}. The expanded query $(Q_0 \cup Q_{PRF})$, is

then used to extract the title and snippet from the top-K web search results. A similar procedure is applied to select the top n_w terms for expansion and referred to as Q_{web}. As hashtag highlights the content of a tweet, we extract the hashtag of top-N retrieved tweets that contain the original query terms for expansion and refer to as $Q_{\#hash}$. Therefore, all expansion terms are appended to the original query as follows:

$$Q_{Expanded} = (Q_0 \cup Q_{PRF}) \cup Q_{web} \cup Q_{\#hash}$$

where $Q_{Expanded}$ is the final expanded query.

4.2.3 Query Type Determination

In microblog, people usually search the recent information when a notable event occurs and tweets that are posted during this period have the similar kind of sentiment. The objective of our query type determination approach is to determine the temporal and sentiment dimensions of the query.

To extract the temporal aspect of queries, we utilize the temporal distribution of the top tweets retrieved by using a baseline method. For example, Figure 4.3 illustrates the temporal distribution of three samples of TREC queries. Based on these distributions, the query can be classified into either time insensitive or time sensitive.

In a time insensitive query, relevant tweets have a relatively flat (uniform) distribution over time, whereas in time sensitive queries, relevant tweets are not spread uniformly over time but rather tend to be concentrated in a certain time period. Time sensitive query usually indicates notable events or issues and may have different temporal patterns. For the query MB001 ("BBC World Service staff cuts"), on January 26, 2011, BBC released a news that they confirmed plans to close five of its 32 World language services and what we see in Figure 4.3 is that a large proportion of relevant tweets posted on January 26, 2011. To clarify this scenario we took another query MB020 ("Taco Bell filling lawsuit"). The relevant tweets of this query are mostly concentrated on two different time periods. Whereas, considering the query MB021 ("Emanuel residency court rulings"), we see that relevant tweets are concentrated on more than two time periods. But the interesting point is the percentage of the relevant tweets are not as high like

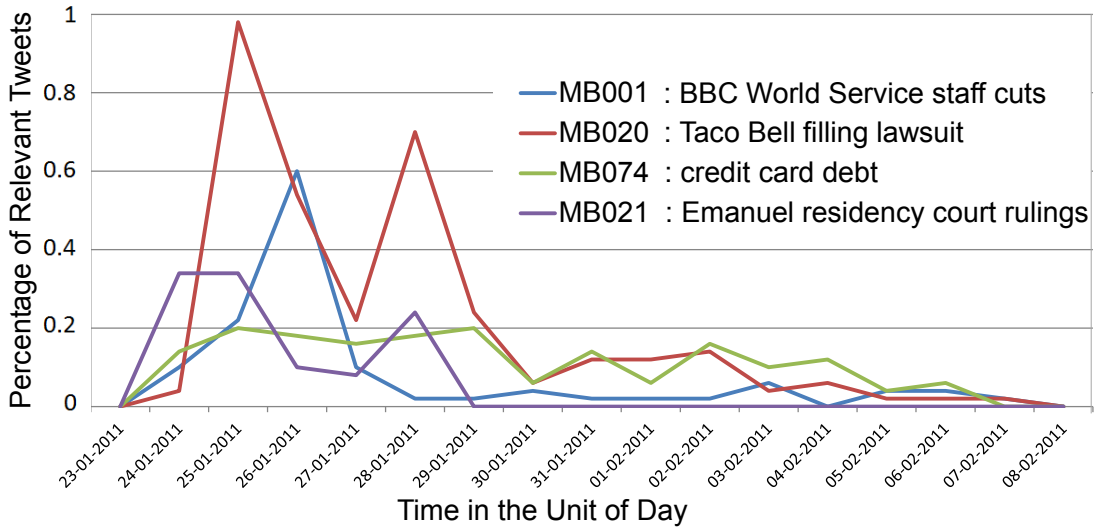

Figure 4.3: Temporal distribution of relevant tweets.

the previous two queries. However, for the query MB074, we see that relevant tweets uniformly distributed over the time. Based on this observation, we again classify the query as dominant-peak query and non-dominant-peak query.

- **Dominant-Peak Queries:**
  In dominant-peak queries, a large number of relevant tweets are concentrated around only one peak and the percentage of the relevant tweets rapidly decreases beyond the peak. As shown in Figure 4.3, query no. MB001 ("BBC World Service staff cuts") is an example of such type of query.

- **Non-Dominant-Peak Queries:**
  In non-dominant-peak queries, relevant tweets are concentrated around more than one peak. Each peak contains a significant portion of relevant tweets, but the percentage is not as high as dominant-peak. As shown in Figure 4.3, query no. MB020 ("Taco Bell filling lawsuit") and MB021 ("Emanuel residency court rulings") are an example of such type of query.

However, in our proposed approach we consider the minimal set of top retrieved tweets based on a query to determine its temporal orientation. Therefore, we consider the hour-wise tweet distribution for timestamp binning process.

Given a query Q with timestamp t, let the top-N tweets retrieved by using the baseline method is $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ and $T = \{t_1, t_2, ..., t_N\}$ be the set of associated publishing timestamp. From this timestamp set, we extract the hour-wise unique timestamp set, $UT = \{h_1, h_2, ..., h_M\}$ that means posts published during the same hour having the same timestamp. Based on the unique timestamp set, UT we estimate the hour-wise tweets distribution, $HT_{\mathcal{D}} = \{(h_1, f_1), ..., (h_M, f_M)\}$, where each pair contains the unique timestamp, h_i with the number of corresponding published tweets, f_i. Then, we estimate the standard deviation from the hour-wise temporal distributions of top-N tweets, $UT_F\{f_1, f_2, ..., f_M\}$.

Finally, for a given query Q, we estimate the temporal sensitivity of the query as follows:

$$f_{TQ}(Q) = \begin{cases} \text{Time sensitive, if } s(UT_F) > P \\ \text{Time insensitive, otherwise} \end{cases} \tag{4.1}$$

where s is the corrected sample standard deviation and P is the threshold value.

Next, we classify the temporal sensitive queries either dominant-peak query or non-dominant peak query. In this regard, we consider only those timestamps which value fluctuates above the standard deviation from the mean. Let, TS be the set of those timestamps and d_{TS} denotes the number of days that the timestamps of TS spanned. Then, we determine the dominant and non-dominant peak query as follows:

$$f_{QueryType}(Q) = \begin{cases} \text{Dominant, if } (d_{TS}) = 1 \\ \text{Non-dominant, if } (d_{TS}) > 1 \end{cases} \tag{4.2}$$

To extract the sentiment aspect of queries, we utilize the sentiment distribution of the top tweets retrieved. We hypothesize that a query is sentiment sensitive, if the largest proportion of the top retrieved tweets have the similar kind of sentiment polarity, including positive, negative, and neutral. For example, in Figure 4.4, we can easily deduce that query no. MB024 and MB036 are sentiment sensitive, because a large proportion of relevant tweets categorized into positive and negative sentiment, respectively. However, query no. MB022 is not sentiment sensitive, where all three categories contain a significant proportion of tweets. Because query MB022 ("healthcare law unconstitutional") is based on a controversial issue and people have diverse sentiments on this topic.

Given a query Q, let $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ be the set of top-N tweets retrieved by using the baseline method and $S = \{s_1, ..., s_N\}$ be the set of associated sentiment polarity. We estimate the sentiment polarity wise tweet distributions, $S_{\mathcal{D}} = \{(c_1, f_1), ..., (c_L, f_L)\}$, where each pair contains the sentiment polarity, c_i with the corresponding number of categorized tweets, f_i. The sentiment sensitivity of the query is estimated as follows:

$$f_{SQ}(Q) = \begin{cases} \text{Sentiment sensitive, if } (S_{max} > S_{th}) \\ \text{Sentiment insensitive, otherwise} \end{cases} \tag{4.3}$$

where S_{th} is the threshold value and

$$S_{max} = \frac{1}{N} \underset{f_i \in S_{\mathcal{D}}}{max}(f_i)$$

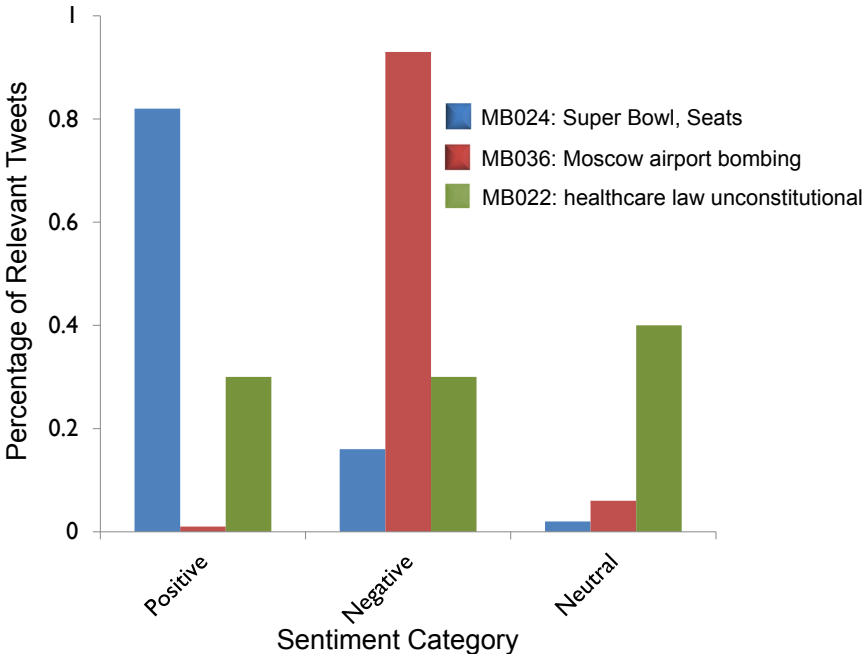

Figure 4.4: Sentiment distribution of relevant tweets.

We extract different feature sets based on the temporal and sentiment sensitivity of queries. If a query is temporally sensitive, we extract the temporal features (i.e., recency score and burst-aware score) described in Section 4.2.4.6, otherwise we exclude these features from the feature set. Similarly, if a query is sentiment sensitive, we extract the sentiment feature described in Section 4.2.4.4, otherwise we exclude this feature from the feature set.

4.2.4 Feature Extraction

For our tweet reranking system, we extract 23 features grouped into 6 different categories. Table 4.1 presents all the features for our tweet learning to rank framework. The feature extraction processes are described in detail as follows:

4.2.4.1 Content Relevance Features

Content relevance feature indicates the lexical similarity between a given query and a target tweet. In our system, we extract six content relevance features, including language model with Dirichlet smoothing [33], TF-IDF [31], Okapi BM25 [34], vector space model [32], divergence from randomness [35], and Jaro-Winkler similarity [36].

4.2.4.2 Twitter Specific Features

Tweets have many special characteristics. We exploit these characteristics and extract some of them as a feature for the ranking model.

Tweet Length (TL): Tweet length means the number of words available in a tweet text. We estimate this feature with the hypothesis that a longer tweet contains more information [11].

URL: To share more vital information, the user usually posts URL on twitter. For instance, the tweet in Figure 4.5[1] contains a URL which leads to a web page that contains detailed information about Microsoft-led research team wins "Marr Prize" for outstanding computer vision research. However, a tweet containing too many URLs might be a spam.

Figure 4.5: A tweet example with URL.

From that perspective, we utilize a URL feature which builds upon the intuition

[1]Image taken from the public twitter post.

Table 4.1: List of features, where our proposed features are highlighted in bold.

Feature Type	Feature Name
Content Relevance Features	1. Language Model with Dirichlet Smoothing [33] 2. TF-IDF [31] 3. Okapi BM25 [34] 4. Vector Space Model [32] 5. Divergence From Randomness [35] 6. Jaro-Winkler Similarity [36]
Twitter Specific Features	1. Tweet Length (TL) [11] 2. URL [169] 3. URL Count (UC) [170] 4. Retweet Count (RTC) [171] 5. Hashtag (HT)
Account Related Features	1. Followers Count (FC) [171] 2. Status Count (SC) [171]
Context Relevance Features	1. **Semantic Language Model (SLM)** 2. **Kernel Density with Language Model (KDLM)** 3. **Kernel Density with Language Model and Recency (KDLMR)** 4. **Sentiment Feature (SF)**
Popularity Based Features	1. Tweet Popularity (TP) 2. **URL Popularity (UP)** 3. **Query Terms in URL (QTU)** 4. **Hashtag Importance (HTI)**
Temporal Featuress	1. **Recency Score (RS)** 2. **Burst-Aware Score (BS)**
Total	23 Features

that the tweet contains a URL has some importance. It is a binary feature that is assigned 1 if a tweet contains at least one URL and 0 otherwise [169].

$$f_{URL}(D) = \begin{cases} 1, \text{ if the tweet } D \text{ contains a URL} \\ 0, \text{ otherwise} \end{cases}$$

URL Count (UC): The UC feature counts the number of URLs published in a tweet D [170]. It is estimated as follows:

$$f_{UC}(D) = |\{u \in D / isURL(u)\}|$$

Retweet Count (RTC): In twitter, informative tweet that is reposted by many users without any modification is called retweet. RTC indicates the number of times a tweet is retweeted. To measure the popularity of a tweet, we use an integer between 0 and 5 (inclusive) based on retweet count [171].

$$f_{RTC}(D) = \begin{cases} 0, \text{ if RTC} = 0 \\ 1, \text{ if RTC} \in [1, 10] \\ 2, \text{ if RTC} \in [11, 100] \\ 3, \text{ if RTC} \in [101, 1000] \\ 4, \text{ if RTC} \in [1001, 10000] \\ 5, \text{ for other values} \end{cases}$$

Hashtag (HT): A hashtag is a type of label or metadata tag used by users within a tweet to highlight a topic on twitter. Our hashtag feature is a binary feature that is assigned 1 if a tweet contains at least one #Hashtag and 0 otherwise.

$$f_{HT}(D) = \begin{cases} 1, \text{ if the tweet } D \text{ contains a \#Hashtag} \\ 0, \text{ otherwise} \end{cases}$$

4.2.4.3 Account Related Features

To estimate the credibility of a tweet author, we extract some account related information for our ranking model.

Followers Count (FC): FC indicates the number of followers that the author of this status has. To measure the credibility of a tweet author, we use an integer between 0 and 5 (inclusive) based on the followers count [171].

$$f_{FC}(D) = \begin{cases} 0, \text{ if FC} = 0 \\ 1, \text{ if FC} \in [1, 10] \\ 2, \text{ if FC} \in [11, 100] \\ 3, \text{ if FC} \in [101, 1000] \\ 4, \text{ if FC} \in [1001, 10000] \\ 5, \text{ for other values} \end{cases}$$

Status Count (SC): SC indicates the number of tweets that the author has already posted before, at the time of posting this tweet. To measure this feature, we use an integer between 0 and 5 (inclusive) based on the status count [171].

$$f_{SC}(D) = \begin{cases} 0, \text{ if SC} = 0 \\ 1, \text{ if SC} \in [1, 10] \\ 2, \text{ if SC} \in [11, 100] \\ 3, \text{ if SC} \in [101, 1000] \\ 4, \text{ if SC} \in [1001, 10000] \\ 5, \text{ for other values} \end{cases}$$

4.2.4.4 Context Relevance Features

The short length characteristics of microblog documents and frequent use of unconventional abbreviations such as "RT" for the retweet and "U" for You, exacerbates the vocabulary mismatch problem during the retrieval process. For alleviating the vocabulary mismatch problem, we propose some context relevance features based on word embedding, kernel density estimation, and query-tweet sentiment correlation.

Semantic Language Model (SLM): Vocabulary mismatch is an obvious challenge in matching query terms with a tweet in microblog retrieval. To overcome the vocabulary mismatch, we first propose a new semantic feature by incorporating the semantic similarity of a query term with a tweet in a language model framework as follows:

$$f_{SLM}(Q, D) = \frac{1}{|Q|} \sum_{q \in Q} \frac{Sim(q, D) + \mu P(q|C)}{|D| + \mu} \tag{4.4}$$

where

$$Sim(q, D) = f_{sim}(\vec{q}, \frac{1}{|D|} \sum_{w \in D} \vec{w})$$

where $\vec{q}$ and $\vec{w}$ are the word vector representations from the *word2vec*[1] model proposed by Mikolov et al. [39], corresponding to words q and w, respectively. The function f_{sim} returns the cosine similarity between two word vectors.

Kernel Density with Language Model (KDLM): To estimate the relevance of a tweet timestamp for a given query Q, we make use of kernel density estimation of each temporal signal and combine it with our proposed semantic language model feature as follows:

$$f_{KDLM}(Q, D) = f_{SLM}(Q|D) \cdot f(t_D)$$

where $f(t_D)$ is estimated by employing the kernel density estimation.

Let $(x_1, x_2, \cdots, x_n)$ be an independent and identically distributed sample drawn from some distribution with an unknown density, f. We are interested in estimating the shape of this function, f. Its kernel density estimator is defined as follows:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big) \tag{4.5}$$

where K($\cdot$) is the kernel, a non-negative function that integrates to one and $h > 0$ is a smoothing parameter called the bandwidth. If Gaussian kernel is used to approximate univariate data, then as Silverman [172] has shown, the optimal choice for bandwidth h is:

$$h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}n^{-1/5}$$

where $\hat{\sigma}$ is the standard deviation of the samples. It is important to note that the choice of a kernel function is mainly a matter of convenience, carrying with it no implications of underlying parametric forms of the data. We select the Gaussian due to its wide use and its ready definition of an optimal bandwidth.

Kernel Density with Language Model and Recency (KDLMR): To emphasize the recent tweets for a given query Q, we combine our recency feature

[1] *word2vec* (https://code.google.com/p/word2vec/)

along with semantic language model (SLM) and kernel density estimation of timestamp as follows:

$$f_{KDLMR}(Q, D) = f_{SLM}(Q|D) \cdot f(t_D) \cdot f_{RS}(Q, D)$$

where the recency function $f_{RS}(Q, D)$ is estimated by using the Eq. (4.7). Function $f(t_D)$ and $f_{SLM}(Q|D)$ are estimated by using the Eqs. (4.5) and (4.4), respectively.

Sentiment Feature (SF): To reward the sentimentally sensitive queries, we propose a sentiment feature (SF) based on query sentiment and tweet sentiment. Our sentiment feature is a binary feature that is assigned 1 if the tweet sentiment and query sentiment are similar and 0 otherwise.

$$f_{SF}(Q, D) = \begin{cases} 1, \text{ if } Q_S = D_S \\ 0, \text{ otherwise} \end{cases}$$

where Q_S denotes the sentiment polarity of the query, Q and D_S denotes the sentiment polarity of the target tweet, D.

4.2.4.5 Popularity Related Features

In microblog, when a notable event occurs, lots of users share similar kinds of posts, URL, and hashtags. That is why; to estimate the importance of a tweet we introduce four popularity based features.

Tweet Popularity (TP): The tweet popularity feature estimates the popularity of a tweet T in the corpus, which is estimated as follows:

$$f_{TP}(Q, D) = \frac{\sum_{D \neq D_i} sim(D, D_i)}{|D_C| - 1}$$

where D_C is the tweet corpus and we make use of cosine similarity to estimate the similarity function $sim(D, D_i)$. In this context, we consider a pair of tweets are similar if their cosine similarity score is greater than 0.5.

URL Popularity (UP): The URL popularity feature estimates the importance of a URL in the corpus, which in turn denotes the importance of a tweet

containing this URL. Our proposed URL popularity feature of a tweet, D is estimated as follows:

$$f_{UP}(D) = e^{log(\sum_{url \in D} D_{url}+1)} \tag{4.6}$$

where D_{URL} is the number of times a URL appear in the corpus.

Query Terms in URL (QTU): In twitter, users usually share URLs in tiny URL format to share extra information. While expanding such tiny URL, we get more insights about what the URL contains. For a given query Q and tweet D, our proposed query terms in URL (QTU) feature is estimated as follows:

$$f_{QTU}(Q, D) = f_{UP}(D) \cdot |\{w \in Q / inURL(w) \in D\}|$$

where $f_{UP}(D)$ is estimated by using Eq. (4.6).

Hashtag Importance (HTI): User generated hashtags are important pieces of information, which generally indicate the trending events or issues. The hashtag importance feature for a given query, Q and tweet, D is estimated as follows:

$$f_{HTI}(Q, D) = \frac{1}{1 + e^{(-1 * \sum_{\{h \in Q \cap \#h \in D\}} IDF(\#h))}}$$

where $IDF()$ is the inverse document frequency.

4.2.4.6 Temporal Features

As microblog posts particularly focus recent news and events, temporal information plays an important role in microblog retrieval. To extract temporal aspect of tweets, we extract our proposed temporal features such as recency score and burst-aware score by utilizing query time and tweet time.

Recency Score (RS): Our recency score feature build upon the intuition that the less time difference between tweet time and query time, the more relevant the query and tweet are. Therefore, we measure the recency score (RS) of a tweet as follows:

$$f_{RS}(Q, D) = \frac{1}{\log_2((Q_T - D_T)^2 + 2)} \tag{4.7}$$

where Q_T denotes the timestamp of the query and D_T denotes the timestamp of the target tweet, D.

Burst-Aware Score (BS): Time sensitive queries have a mostly uneven distribution of relevant tweets over time and in Section 4.2.3, we categorize them as dominant-peak query and nondominant-peak query. Burst, defined as "*a brief period of intensive activity followed by long period of nothingness*" [173], is a common phenomenon in human activities. The bursty nature of human behavior is observed and studied extensively in many domains. Kleinberg et al. [174], proposed a weighted-automaton model to discover the bursty and hierarchical structure in document streams of email and news articles. Amodeo et al. [128] detected bursts for timed query expansion using Rocchio's pseudo relevance feed-

Algorithm 1: BurstDetector: An algorithm for detecting burst timestamp in microblog posts (tweet).

Input: A list of tweets, $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ with corresponding list of timestamp, $T = \{t_1, t_2, ..., t_N\}$

Output: Burst Timestamp, B_T

Burst Timestamp, $B_T = \varnothing$
Burst Timestamp Candidates, $C_T = \varnothing$
Histogram, $H_T = getTimestampHistogram(\mathcal{D}, T)$
$T_{sd} = s(H_T)$
$\bar{f} = getAverage(H_T)$
for $h_i \in H_T$ **do**
    $f_i = getFrequency(H_T, h_i)$
    if $(f_i > (T_{sd} + \bar{f}))$ **then**
        $PutPair(C_T, (h_i, f_i))$
$d_s = getDaysSpanned(C_T)$
$queryType = getQueryType(d_s)$
if $(queryType == \text{"}Dominant\text{"} || (d_s == 2))$ **then**
    for $h_i \in C_T$ **do**
        $B_T = B_T \cup h_i$
else
    $B_T = B_T \cup h_i$ where $h_i = \underset{f_i}{argmax}((h_i, f_i) \in C_T)$
return B_T

back. More recently, Rao et al. [18] utilized the continuous hidden markov model (cHMM) to identify documents that occur in bursty temporal clusters.

Our burst aware score feature reward tweets, occur in the bursty time state. To detect the burst timestamp, we propose an Algorithm 1, where the input is a set of top $N = 30$ tweets, $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ with a corresponding set of timestamp, $T = \{t_1, t_2, ..., t_N\}$, and output is a burst timestamp set, B_T. At first, we initialize the burst timestamp set, B_T and burst timestamp candidates, C_T. Next, we compute the histogram of timestamp, H_T. The $getTimestampHistogram()$ function returns the hour basis unique timestamps set with the corresponding number of posted tweets in these timestamps. After that, we estimate the corrected sample standard deviation, s based on the frequency of H_T i.e. number of posted tweets. Next, those timestamps which value fluctuates above the standard deviation from the mean, we put them in burst timestamp candidates, C_T. The $getDaysSpanned()$ function returns the number of days that the timestamps of C_T spanned. Then, the $getQueryType()$ function estimate the type of the query by using Eq. 4.2. Finally, we estimate the burst timestamp set, B_T based on the condition that, if the query type is "Dominant" or the burst timestamp candidates, C_T are spanned in two days, then we consider all the timestamps as the bursts whereas the timestamp that has the highest number of posted tweets is considered as a burst in the other cases.

Based on the burst timestamp set, B_T, we determine the single burst timestamp, B_{ST} of a tweet, D as follows:

$$B_{ST}(D, B_T) = nearestBurst(D_T, B_T)$$

where the $nearestBurst()$ function returns the timestamp of B_T set that has the minimum Euclidean distance with respect to tweet timestamp, D_T.

Finally, we measure the burst-aware score (BS) of a tweet, D as follows:

$$f_{BS}(D) = \frac{1}{\sqrt{|B_{ST} - D_T| + 1}}$$

where B_{ST} denotes the burst timestamp of the tweet and D_T denotes the timestamp of the target tweet.

4.2.5 Supervised Feature Selection

To improve the performance of our tweet reranker system, we make use of elastic-net regularized regression method [175], a well-known supervised feature selection (SFS) approach that selects the best feature combination by eliminating irrelevant features. With a positive regularization parameter λ, elastic-net minimizes the following objective function:

$$f_{ElasticNet} = \min_{\beta_0 - \beta} \left(\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda \sum_{j=1}^{P} \left(\frac{(1-\alpha)}{2} \beta_j^2 + \alpha |\beta_j| \right) \right)$$

where N is the number of observations, y_i is the response of observations i, x_i is the data. The elastic-net penalty is controlled by α, which is strictly between 0 and 1. We train our elastic-net model based on each observation as a query-tweet pair and select those features as relevant which have the positive coefficient β.

4.2.6 Ranking Model

To estimate the relevance score for tweet reranking, we design a linear learning to rank (L2R) model. Given a query Q and a tweet document D, the relevance score value, rsv is estimated as follows:

$$rsv(Q, D) = \frac{\sum_{i=1}^{N} \lambda_i f_i(Q, D)}{\sum_{i=1}^{N} \lambda_i} \tag{4.8}$$

where N is the number of features, $f_i(Q, D)$ is a feature function, and λ_i is a model parameter.

To instantiate the model parameter λ_i stated in Eq. (4.8), we make use of two state-of-the-art learning to rank models such as random forest and SVM^{rank} [176]. Next, reciprocal rank fusion (RRF) method [177] is applied to combine the results from these ranking models. RRF sorts the documents according to a naive scoring formula. Given a set $\mathcal{D}$ of documents to be ranked and a set of rankings R, each a permutation on $1 \cdots |\mathcal{D}|$, RRF_{score} is estimated as follows:

$$RRF_{score}(D \in \mathcal{D}) = \sum_{r \in R} \frac{1}{k + r(D)} \tag{4.9}$$

The constant k mitigates the impact of high rankings by outlier systems.

4.3 Experiments and Evaluation

4.3.1 Experimental Setup

Dataset Collection: In order to evaluate our tweet reranker system, we make use of Tweets2011 corpus used in the TREC microblog 2011 (TMB2011) and 2012 (TMB2012) tracks. The collection consists of approximately 16 million tweets. We used the official TREC microblog search API [115] for retrieving 1000 tweets using the baseline method. The official query topics used in the TMB 2011 and TMB 2012 were consisted of 49 (TMB2011) and 60 (TMB2012) timestamped topics. TREC also provided the relevance judgments of tweets for these query topics. There are three relevance levels, including irrelevant (labeled 0), minimally relevant (labeled 1), and highly relevant (labeled 2). We evaluated our proposed method in ranking tweets in descending order of relevance for both *allrel* and *highrel* criteria. *Allrel* considers both minimally and highly relevant tweets as relevant, whereas *highrel* only considers the highly relevant tweets as relevant. As depicted in Figure 4.6, each topic is composed of query_id, query_text, query_time; while each tweet document (depicted in Figure 4.7) is composed of tweet_id, screen_name, tweet_time, tweet_text, followers_count, statuses_count, retweeted_count, etc. TREC Search API provided ranking results by using Lucene's implementation of query-likelihood (LMDirichletSimilarity), which we considered as our *baseline*. To extract context relevance features based on word embedding, we trained 400-dimensional *word2vec* model on Tweets2011 corpus and used the word vectors accordingly.

< *top* >
< *num* > Number: MB088 < /*num* >
< *query* > Kings' Speech awards < /*query* >
< *querytime* > Tue Feb 08 00:48:24 +0000 2011 < /*querytime* >
< *querytweettime* > 34775520600129536 < /*querytweettime* >
< /*top* >

Figure 4.6: Sample query.

< *item* >

< *id* > 31933013126287360 < /*id* >

< *rsv* > 7.848692893981934 < /*rsv* >

< *screen_name* > kymlewison < /*screen_name* >

< *epoch* > 1296448398 < /*epoch* >

< *text* > #KingsSpeech The King's Speech wins at SAG Awards:
The King's Speech wins the best-actor trophy Sunday for...
http://bit.ly/hcaOIvbythere < /*text* >

< *followers_count* > 10 < /*followers_count* >

< *statuses*$_c$*ount* > 7170 < *statuses_count* >

< *retweeted_count* > 740 < /*retweeted_count* >

< /*item* >

Figure 4.7: Sample tweet.

Results with Supervised Feature Selection: For supervised feature selection by using elastic-net regularization method, we applied a publicly available package glmnet [178]. The result of our supervised feature selection process indicates that *Divergence from Randomness*, *Hashtag*, and *Followers Count* features are irrelevant. Here, we describe our interpretation behind this selection.

#Hashtag Feature: Our proposed #hashtag feature was a simple binary feature, which is assigned 1 if a #hashtag is found in a tweet documents and 0 otherwise. We didn't consider some vital information about #hashtag including #hashtag statistics, #hashtag segmentation, #hashtag popularity over the corpus etc. That is why; we think that our #hashtag feature is not selected as relevant.

Followers Count Feature: Followers count may not be a good feature for tweet relevancy measure. In Microblog information retrieval, it is not necessary to estimate how many peoples followed you; rather it is necessary to know how many people discuss about the query topic. We need not follow a user to search or retweet his posted tweets. Moreover, a large number of twitter users turn in

twitter site when a notable event occurs. That is why; we think that our followers count feature is not selected as relevant. *Divergence from Randomness Feature:* Divergence from randomness (DFR) models build upon the intuition that the more the content of a tweet document diverges from a random distribution, the more informative the tweet is. But when a notable event occur a large number of twitter users widely discuss about this topics. As they discuss about a specific topic or events their discussions contains seemingly similar kinds of contents that means less diversification. So, model that emphasize on diversification might played a negative role here. Hence, we think that our divergence from randomness feature is not selected as relevant.

Feature Importance Estimation: In order to estimate the importance of our automatically selected features, we make use of a publicly available package of random forest [179]. We utilize this package to estimate the *MeanDecreaseGini*, a measure of variable importance in random forest model. Every time a split of a node is made on feature f, the *Gini* impurity criterion for the two descendent nodes is less than the parent node. Adding up the *Gini* decreases for each individual feature over all trees in the forest gives an importance score of each feature [180]. Ranked list of our selected features based on importance score is illustrated in Figure 4.8, where proposed features are highlighted in boldface. Among all the 20 selected features, our proposed temporal features were ranked at second and fourth position, which denotes the complementary importance of temporal features. Therefore, combining temporal features with other features achieved enhanced performance. Along with this direction, our proposed context relevance features were ranked at eighth, ninth, tenth, and fifteenth position, whereas our popularity features were ranked at seventh, thirteenth, and nineteenth position, respectively. From this observation, we can deduce that our proposed features are effective for tweet reranking.

Training and Testing L2R Model: For our linear learning to rank model stated in Eq. (4.8), we make use of publicly available packages of random forest [179] with no parameter tuning. Feature importance scores (*MeanDecreaseGini*) of our selected features obtained from the random forest are used to instantiate the model parameter, λ_i. We denote this setting as *LWL2RRF*. We

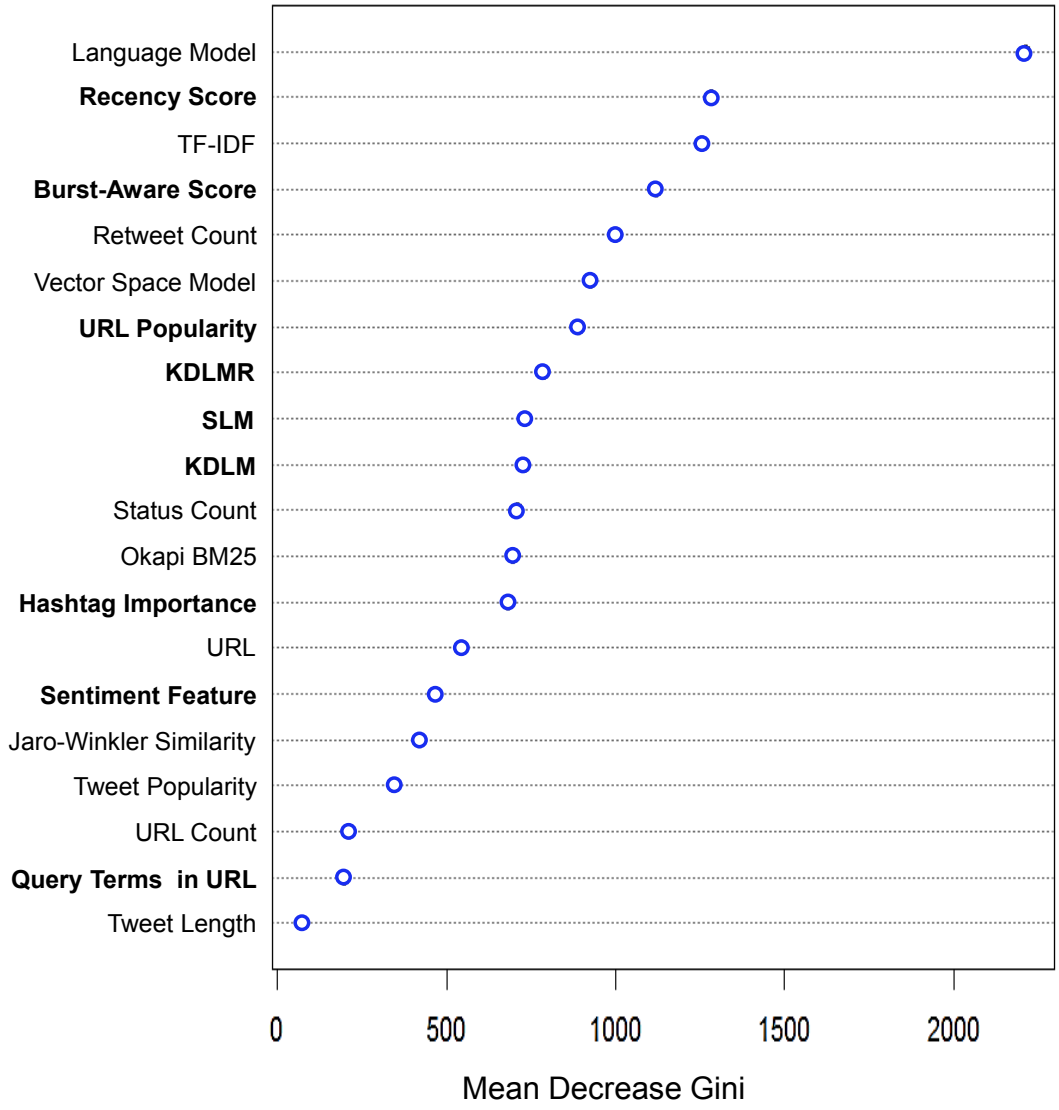

Figure 4.8: Feature importance.

also employ SVM^{rank} [176], a state-of-the-art learning to rank model based on our selected features. We denote this setting as *LWL2RSVM*. In both settings, at first we train on TMB2011 topics and test on TMB2012 topics, and vice versa.

Parameter Setting: For PRF (Q_{PRF}) and hashtag ($Q_{\#hash}$) based query expansion, we utilized the top-N tweets retrieved by the baseline method. We set N to 30, because of Miyanishi et al. [19] reported that when N is large ($N > 30$),

the performance is not sensitive to the choice of N. To select the optimal number of feedback terms in PRF, we performed the grid search based on both TMB2011 and TMB2012 test collections. The optimal number of feedback terms was set as $n_p = 3$. For web-search (Q_{web}) based query expansion, we empirically used $K = 16$ search results and the optimal number of feedback terms was set as $n_w = 10$. Later, our query expansion strategy is applied to combine them.

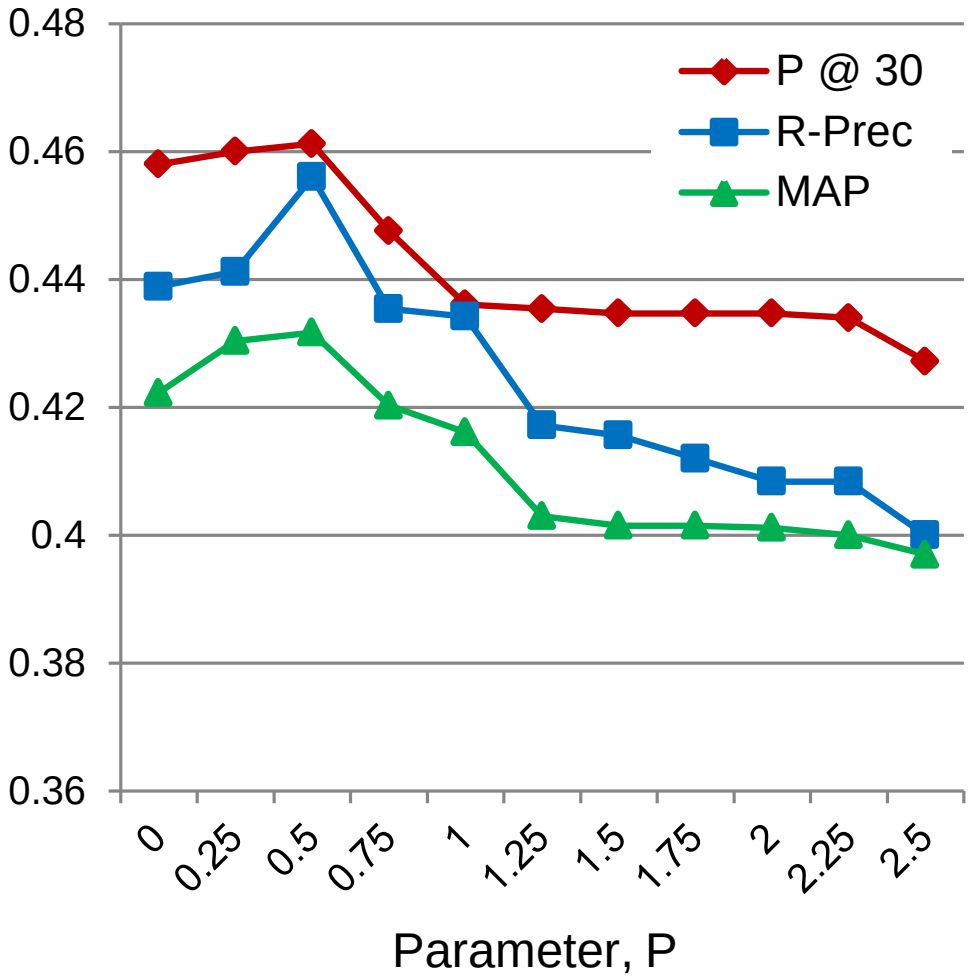

Figure 4.9: Sensitivity of paramter, P in Eq. (4.1).

To determine the optimal value of parameter P in Eq. (4.1), we utilize the top-N tweets retrieved by the baseline method. We set N to 30. Next, we examine the performance of our method *LWL2RRF* for different values of P by utilizing the TMB2011 and TMB2012 test collections, where we only consider the

Language Model with Dirichlet Smoothing and temporal features. In both cases, we got the nearly similar kind of performances. For instance, the result based on TMB2011 test collection is illustrated in Figure 4.9. It is observed that when the value of the parameter P is 0.5, we obtained the best result in terms of all three evaluation measures and the parameter P is set as 0.5.

To estimate the sentiment of each tweet, we applied a publicly available package SentiStrength [181]. The optimal value of parameter S_{th} in Eq. (4.3) is set to as $S_{th} = 0.7$. We performed the grid search based on both TMB2011 and TMB2012 test collections to estimate this optimal value.

We set the constant, k in Eq. (4.9) as 60, according to the recommendation by Cormack et al. [177].

4.3.2 Evaluation Measures

To evaluate the effectiveness of our proposed micrblog reranker method, i.e. the proportion of correctly retrieved relevant tweets for a given query, we used four standard information retrieval (IR) evaluation measures, including precision at top 30 tweets (P@30), mean average precision (MAP), reciprocal-precision (R-Precision) [6, 182], and normalized discounted cumulative gain at top 30 tweets (NDCG@30).

Precision at Rank K (P@K):
In the field of information retrieval, precision is the fraction of the retrieved tweets that are relevant to a given query [24]. The formula to estimate the precision is defined as follows:

$$\text{Precision, } P = \frac{|\{\text{relevant tweets}\} \bigcap \{\text{retrieved tweets}\}|}{|\{\text{retrieved tweets}\}|}$$

Along with this direction, precision at rank K i.e. P@K measures the proportion of good results among the first K number of retrieved tweets. According to the TREC microblog benchmark [6], we consider K=30 i.e. precision at rank 30 (P@30) as one of the evaluation measure. This evaluation measure is important for the microblog search, since users tend to look at only the top ranked resulting tweets.

Mean Average Precision (MAP):

Another important evaluation measure that we used is the mean average precision (MAP) [24], which captures both the precision and recall. MAP for a set of queries is the mean of the average precision scores for each query. MAP provides a single-figure measure of quality across recall levels. Therefore, among the evaluation measures, MAP has been demonstrated the good discriminative power and stability. For a single query, average precision (AP) is the average of the precision value obtained for the set of top-k tweets existing after each relevant tweet is retrieved, and this value is then averaged over all the queries. That is, if the set of relevant tweets for a query is $q_j \in Q$ is $\{d_1,.....d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until we get to tweet d_k, then MAP is estimated as follows:

$$\text{MAP (Q)} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j}\text{Precision } (R_{jk})$$

where Q is the number of queries.

Since the average is over all the relevant tweets, therefore the precision value in the above equation will be 0 when the system didn't retrieved any relevant tweet. For a single query, the AP approximates the area under the un-interpolated precision-recall curve, therefore for a set of queries the MAP is roughly the average area under the precision-recall curve.

R-Precision:

R-Precision is the precision after R tweets that have been retrieved, where R is the total number of relevant tweets for the query [183]. When considering a set of relevant tweets, R-Precision refers to the best precision on the precision curve. However, it de-emphasizes the exact ranking of the retrieved relevant tweets, which is important for our microblog retrieval tasks since it contains three relevance levels.

Normalized Discounted Cumulative Gain (NDCG):

The normalized discounted cumulative gain (NDCG) is a widely used evaluation measure to estimate the effectiveness of a reranker system. NDCG is basically designed for ranking tasks that consider more than one relevance levels. Since our

TREC microblog dataset [6] contains three relevance levels (irrelevant, relevant, and highly relevant), we consider NDCG as one of the evaluation measure to estimate the performance.

Discounted cumulative gain (DCG) is the predecessor of NDCG. The premise of DCG is that highly relevant tweets appearing lower in a ranked result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result.

The traditional formula of DCG accumulated at a particular rank position K is defined as follows [184]:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{rel_i}{log_2(i+1)}$$

An alternative formulation of DCG [185] places stronger emphasis on retrieving more relevant tweets:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

Currently, this later formula is very popular among the major web search companies and data science competition platforms.

However, search result lists might be vary in length depending on the type of the query. Therefore, only using DCG cannot consistently compare the performance of retrieval system from one query to the next. That is why; the cumulative gain at each position for a chosen value of K should be normalized across queries. This is done by sorting all relevant tweets in the corpus by their relative relevance, producing the maximum possible DCG through position K, also called Ideal DCG (IDCG) through that position. The normalized discounted cumulative gain, NDCG, is then estimated as follows:

$$\text{NDCG@K} = \frac{\text{DCG@K}}{\text{IDCG@K}}$$

where

$$\text{IDCG@K} = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

and $|REL|$ denotes the list of relevant tweets ordered by their relevance in the corpus up to position K.

4.3.3 Results with Reranking

To evaluate the performance of our tweet reranking system, we applied four evaluation measures described in Section 4.3.2, including precision at top 30 tweets (P@30), mean average precision (MAP), normalized discounted cumulative gain at top 30 tweets (NDCG@30), and R-Precision (R-Prec). P@30 was used as the official measurement in the TREC microblog ad-hoc search task [6]. We consider this as the primary evaluation measure. For statistical significance testing between two runs' performances, we used a two sided paired t-test at 95% confidence level, where † denotes the statistically significant at ($p < 0.05$), + denotes the moderately significant at ($0.05 \leq p \leq 0.1$), and $\diamond$ denotes the statistically indistinguishable.

In Table 4.2 and Table 4.3, the summarized results of our experiments are presented. At first, we showed the reranking performance based on baseline, which is Lucene's implementation of query-likelihood (LMDirichletSimilarity) model. Results based on two different learning to rank settings were presented in *L2RRF* and *L2RSVM*, respectively. In *L2RRF* setting, the feature importance scores (*MeanDecreaseGini*) obtained from the random forest (RF) were used to instantiate the model parameter λ_i stated in Eq. (4.8). In *L2RSVM* setting, we make use of publicly available SVM^{rank} [176] with default parameter settings. Results of both *L2RRF* and *L2RSVM* settings were combined in *EnL2R* setting by using a reciprocal rank fusion method, stated in Eq. (4.9).

Table 4.2: Performance (P@30, R-Prec, MAP, and NDCG@30; higher is better) on TMB2011 queries for various experimental settings. The best results are highlighted in boldface. † indicates statistically significant difference from the baseline (two sided paired t-tests: $p < 0.05$).

Method	Allrel				Highrel		
	P@30	R-Prec	MAP	NDCG@30	P@30	R-Prec	MAP
Baseline	0.3483	0.3509	0.3050	0.4374	0.1253	0.2405	0.2378
L2RSVM	0.5238†	0.5011†	0.4915†	0.6186†	**0.1899**†	**0.3375**†	0.3339†
L2RRF	**0.5333**†	0.5113†	0.5015†	0.6298†	0.1859†	0.3313†	0.3404†
EnL2R	0.5327†	**0.5198**†	**0.5088**†	**0.6304**†	0.1869†	0.3269†	**0.3442**†

Table 4.3: Performance (P@30, R-Prec, MAP, and NDCG@30; higher is better) on TMB2012 queries for various experimental settings. Legend settings are identical to Table 4.2.

Method	Allrel				Highrel		
	P@30	R-Prec	MAP	NDCG@30	P@30	R-Prec	MAP
Baseline	0.2932	0.2354	0.1815	0.2862	0.1542	0.1751	0.1318
L2RSVM	0.4684†	0.3710†	0.3214†	0.4362†	0.2469†	0.2354†	0.2167†
L2RRF	**0.4729**†	0.3716†	0.3233†	**0.4385**†	**0.2497**†	**0.2405**†	0.2171†
EnL2R	0.4706†	**0.3719**†	**0.3247**†	0.4371†	0.2475†	0.2391†	**0.2176**†

Results showed that all three methods, the *L2RSVM*, the *L2RRF*, and the *EnL2R* significantly ($p < 0.05$) outperforms the baseline for both *allrel* and *highrel* relevant criteria in terms of all evaluation measures on both TMB2011 and TMB2012 queries. This observation validates the effectiveness of our proposed features and techniques to improve the performance of microblog retrieval.

To show the effectiveness of our query expansion technique, we presented the performance of our proposed *EnL2R* method with and without query expansion (QE) in Table 4.4. Results showed that, excluding query expansion, the performance decrease significantly ($p < 0.05$) for *allrel* criteria in terms of all evaluation measures, which in turns deduce the importance of our query expansion technique in microblog retrieval.

Figure 4.10 and Figure 4.11 illustrates the query-wise performance of our proposed *EnL2R* method for *allrel* relevant criteria based on individual test queries of TMB2011 and TMB2012 query set. It shows that P@30 values varied widely across all the queries. Our system obtained P@30 values lower than 0.1 on 6 queries of 2011 query set and 5 queries of 2012 query set. Further examination revealed that these worst queries had very few (5 or 7) relevant tweets in relevance judgment and each relevant tweet rarely contains original query terms which made them difficult to retrieve.

Table 4.4: Performance comparison of our method with/without query expansion (QE) on TMB2011 and TMB2012 test collections. The best results are highlighted in boldface. † indicates statistically significant difference from the method without QE and ⋄ indicates statistically indistinguishable (two sided paired t-tests: $p < 0.05$).

Method	Allrel				Highrel		
	P@30	R-Prec	MAP	NDCG@30	P@30	R-Prec	MAP
EnL2R (TMB2011)	**0.5327**†	**0.5198**†	**0.5088**†	**0.6304**†	**0.1869**⋄	**0.3269**⋄	**0.3442**⋄
Without QE	0.4930	0.4750	0.4547	0.5765	0.1818	0.3186	0.3283
EnL2R (TMB2012)	**0.4706**†	**0.3719**†	**0.3247**†	**0.4371**†	**0.2475**†	**0.2391**⋄	**0.2176**†
Without QE	0.4094	0.3236	0.2870	0.3827	0.2141	0.2236	0.1882

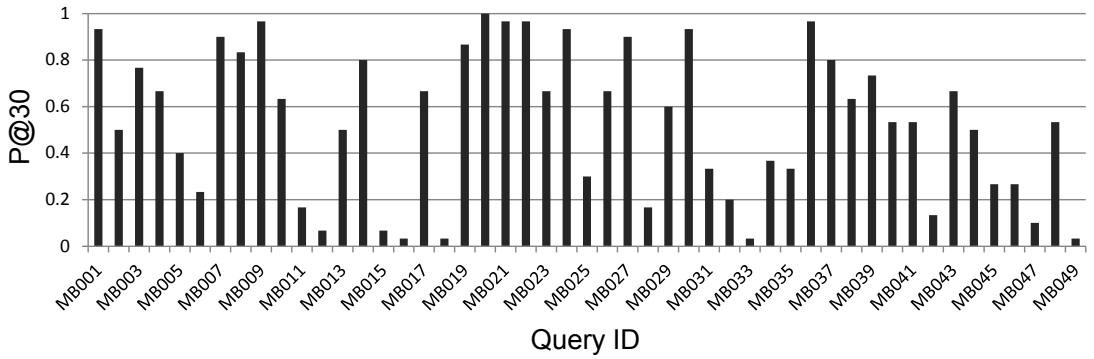

Figure 4.10: Query-wise performance analysis (TMB2011 query set).

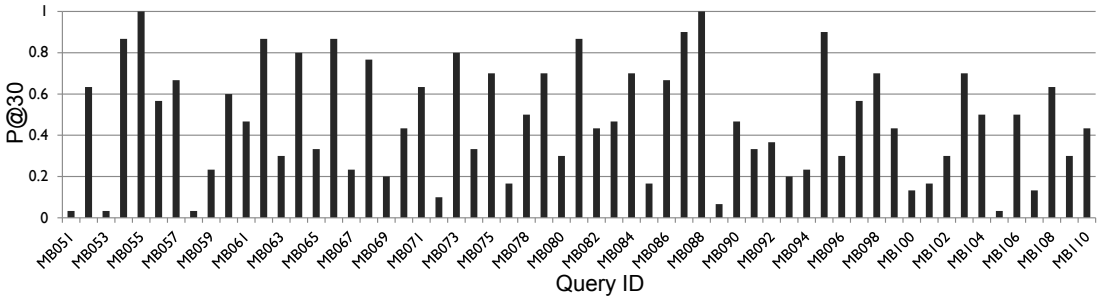

Figure 4.11: Query-wise performance analysis (TMB2012 query set).

4.3.4 Feature Analysis

To understand the effectiveness of our several proposed features and techniques, we divided them into 5 groups, including temporal features, context relevance features, popularity features, query type determination technique, and query expansion technique. We evaluated the effectiveness of each group with a feature ablation study by utilizing TMB2011 test collection, that means removing one group each time and repeated the experiment. Results of these experiments are illustrated in Figure 4.12

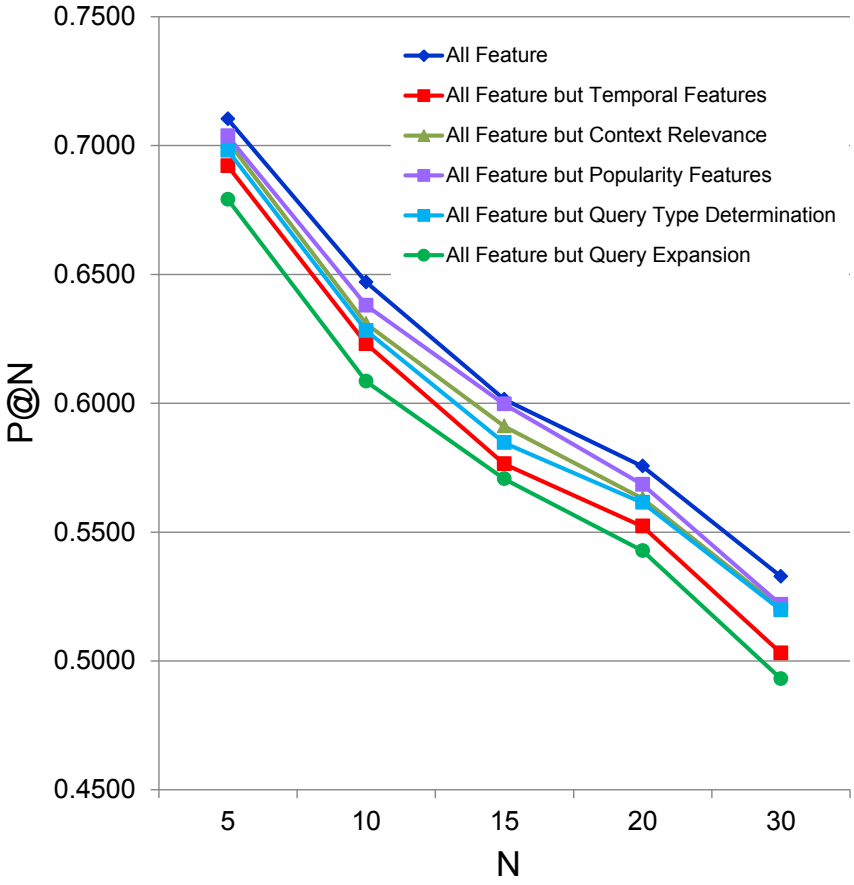

Figure 4.12: P@N performance with different proposed features.

In Figure 4.12, it can be observed that P@N drops substantially, when removing temporal features and the difference in results is statistically significant ($p < 0.05$). This deduced the importance of our temporal features in microblog retrieval. Similar things happened while removing the query-type determina-

tion feature, which revealed the importance of understanding queries underlying temporal and sentiment sensitivity, even though the difference is moderately significant ($0.05 \leq p \leq 0.1$). Removing query-expansion feature would also lead to a significant ($p < 0.05$) decrease in precision, which deduced the importance of query expansion. Popularity features and context relevance features seem to be less important in comparison to temporal features and query expansion feature. But the decrease in performance is moderately significant ($0.05 \leq p \leq 0.1$) while removing these features, thus deduced the importance of these features in microblog retrieval.

4.3.5 Comparison with Related Work

We compared the performance of our proposed method with the known related works [14, 18, 165, 19, 12, 164, 13, 186, 130]. The comparative results of our proposed method with the known related works of TREC Microblog 2011 test collection are described in Table 4.5. Significance testing was conducted with other related methods for comparison but [14] (omitted due to the unavailability of results file and the limitation of reproducing accurate results). For *allrel* criteria, significant differences were observed in our method compared to [19, 12, 164, 13, 18], and baseline in terms of all evaluation measures. For *highrel* criteria, significant differences were observed in [12, 164] and baseline in terms of p@30 and in [12, 164, 13] and baseline in terms of map.

Jia et al. [14] addressed the structural difference and temporality of tweets. Their methods relied on external resources (explore the web pages while URL's are available in tweets), which helped their method to achieve the good performance. However, we obtained a competitive performance without using such external resources and our temporal aspect based features are different from them. Moreover, their method lacks of query expansion, which has been shown effective in our experiments. Miyanishi et al. [19] proposed a time-based query expansion (QE) method to handle the recency and temporal variation. However, following the temporal dimension of the query, our proposed recency and burst-aware features effectively addressed the temporal variation of the tweets. Metzler et al. [12]

Table 4.5: Comparative results with other methods on TMB2011. † indicates the statistically significant difference between our method and the other methods; ⋄ indicates statistically indistinguishable (two sided paired t-tests: $p < 0.05$).

Method	Allrel			Highrel		Paired T-test
	P@30	MAP	NDCG@30	P@30	MAP	
Our Method	**0.5327**	0.5088	**0.6304**	0.1869	0.3442	
Jia et al. [14]	0.5218	**0.5270**	0.5076	**0.2283**	**0.4357**	N/A
Miyanishi et al. [19]	0.4830†	0.2741†	-	-	-	
Metzler et al. [12]	0.4551†	0.2210†	0.4922†	0.1434†	0.1582†	
Amati et al. [164]	0.4401†	0.2318†	0.5086†	0.1495†	0.2048†	
Rao et al. [18]	0.4388†	0.4024†	-	-	-	
Liang et al. [13]	0.4177†	0.2365†	-	0.1979⋄	0.2722†	
Baseline	0.3483†	0.3050†	0.4374†	0.1333†	0.2518†	

proposed a method where they utilized pseudo-relevance feedback via latent concept expansion to address the vocabulary mismatch problem and a number of features with an L2R model to quantify the quality of microblog content. Along with this direction, Amati et al. [164] introduced a Kullback-Leibler based product of information measures (KLIM) model for IR and used the out-of-the-box parameter free query expansion methodology of terrier. However, our three-stage query expansion technique and contextual features effectively addressed the vocabulary mismatch problem. Moreover, we utilized a rich set of account related, twitter-specific, and popularity-based features to quantify the document quality. Liang et al. [13] proposed several temporal features for tweet reranking and Rao et al. [18] utilized the continuous hidden Markov model (cHMM) to identify documents for query expansion that occur in bursty temporal clusters. However, they didn't estimate the temporal dimension of the query, though some queries are temporally insensitive.

In Table 4.6, we described the comparative results of our proposed method with the known related works on TREC Microblog 2012 test collection. Significance testing was conducted with other related methods for comparison but [14, 130] (omitted due to the unavailability of results file and the limitation of reproducing accurate results). For *allrel* criteria, significant differences were observed

Table 4.6: Comparative results with other methods on TMB2012. Legend settings are identical to Table 4.5.

Method	Allrel			Highrel		Paired T-test
	P@30	MAP	NDCG@30	P@30	MAP	
Our Method	**0.4706**	0.3247	0.4371	0.2475	0.2176	
Han et al. [186]	0.4695◇	**0.3469**◇	**0.4625**◇	0.2701◇	0.2642†	
Jia et al. [14]	0.4695	0.3415	0.3018	**0.2738**	**0.2719**	N/A
Fan et al. [130]	0.4611	0.3180	-	-	-	N/A
Liang et al. [165]	0.4062†	0.2786†	0.4240◇	0.2333◇	0.2263◇	
Rao et al. [18]	0.3514†	0.2325†	-	-	-	
Baseline	0.2932†	0.1815†	0.2862†	0.1542†	0.1318†	

in our method compared to [165, 18], and baseline in terms of P@30 and map. We obtained a competitive performance in NDCG@30, although our result is statistically indistinguishable with related methods [186, 165]. For *highrel* criteria, a significant difference is observed in comparison to baseline. However, in terms of p@30 and map, our performance is competitive with related methods [186, 165].

Han et al. [186] utilized the information from webpages whose URL's embedded in tweets. Their method based on query expansion, tweet expansion, a set of textual, non-textual, and user related features with a learning to rank framework. However, they didn't address the temporal and contextual aspect of tweets. Fan et al. [130] proposed a feedback entity model and integrated it into an adaptive language modeling framework to overcome the vocabulary mismatch problem. However, we alleviated the vocabulary mismatch problem by utilizing our three-stage query expansion technique and contextual features. Liang et al. [165] utilized several proposed features including semantic features, tweet related features, and temporal features in an L2R framework. However, they didn't address the temporal dimension of the queries, though some queries are temporally insensitive.

4.3.6 Discussion

As a specific example, we took a query topic MB010 "Egyptian protesters attack museum" and lists the top 10 tweets ranked by our method, Metzler et al. [12] (achieved the best performance in TREC-2011 [6]), Amati et al. [164], and baseline in Table 4.7. ► indicates the relevant tweets with relevance level (2:highly relevant, 1:relevant) and • indicates the non-relevant tweets. It shows that our method returned 9 highly relevant tweets among the top 10 tweets, whereas baseline and Metzler et al. [12] returned only 3 highly relevant tweets. Amati et al. [164] returned 3 highly relevant tweets and 1 relevant tweets. Both baseline and Amati et al. [164] returned a number of retweets, which seems to be relevant. But according to TREC relevance judgments [6], retweets are treated as irrelevant.

Our observation revealed that methods that use external resources (e.g., exploring the web pages using the embedded URL's) achieved better performances in *highrel* criteria, which in turns beneficial while evaluating their methods in *allrel* criteria too. While considering a real-time system, utilizing such external resources is computationally cumbersome and time-consuming. However, we achieved the competitive performance in comparison with these methods [14, 186] without using such external resources and significantly outperformed the other related methods [19, 12, 13, 164, 18, 165].

Moreover, we also demonstrated the effectiveness of our reranker method by taking a sample query "White Stripes breakup". In this regard, we compared the rank of the retrieved tweets by our reranker method with their baseline rank. The results are demonstrated in Table 4.8. It showed that the baseline rank of these relevant tweets (e.g. rank 351, 329, 391, 397 and so on) are far from the top positions and our reranker method ranked these relevant tweets successfully. However, for the query "British Government cuts" (retrieved results are depicted in Table 4.9), we see that the first relevant tweet appeared at the position 8th and there is only one relevant tweet among the top 10 tweets. Further observation revealed that there is very few relevant tweets for this query and other methods also failed to retrieve the relevant tweets for such worst queries.

Table 4.7: Ranked list of top 10 tweets for the query topic MB010, "Egyptian protesters attack museum". URL's are replaced with the word "URL".

Methods	Ranked Tweets
Our Method	1. Looters destroy mummies in Egyptian Museum : official URL #Jan25 #Egypt #daddies ▶ 2 2. Damaged artifacts from Egyptian National Museum #3 (Caps from Al Jazeera Live Stream) #egypt #jan25 URL ▶ 2 3. Artifacts have been stolen from the The Egyptian Museum in Cairo, so sad. URL ▶ 2 4. Looters destroy mummies in Egyptian Museum URL ▶ 2 5. Egypt army secures museum with pharaonic treasures: report: CAIRO (Reuters) - Army units secured the Egyptian URL ▶ 2 6. Looters destroy mummies in Egyptian Museum: official URL ▶ 2 7. Egyptian protests intensify; demonstrators battle with police URL • 8. Gamble: Does the world care more about the Egyptian Museum's artifacts or the freedom of 80m people? URL #Egypt ▶ 2 9. Rioters destroy two mummies in Egyptian Museum in Cairo: Filed under: Arts and Culture, History, Learning, URL ▶ 2 10. @cgorman "at National Museum... protesters formed a "human shield" around the museum to defend from possible looting..." URL ▶ 2

Continued on next page

Table 4.7 – continued from previous page

Methods	Ranked Tweets
Metzler et al. [12]	1. Egyptian protesters return to the streets URL • 2. Egyptian Protester Shot....video URL • 3. "@AmrEldib: "A Very Touching Story" about protesters protecting the Museum URL #Jan25 #Egypt "► 2 4. al Jazeera: Thousands of Egyptian youth form human shields to protect the Egyptian Museum. #Museum #Egypt #Jan25 ► 2 5. Egyptian protesters feel world has passed them by - Washington Post: Telegraph.co.uk Egyptian protesters feel URL • 6. Gamble: Does the world care more about the Egyptian Museum's artifacts or the freedom of 80m people? URL #Egypt ► 2 7. At Egyptian Embassy London Thousands of people,cheering in solidarity with Egyptian protesters.Good energy.solidarity • 8. Check this video out – Police arresting and beating Egyptian protesters URL via @youtube #Jan25 • 9. Egypt police, protesters clash for second day (AFP): AFP - Egyptian police and protesters clashed in the ce... URL • 10. #egypt Egyptian protesters march, denounce Mubarak: Thousands of anti-government protesters have broken a ... URL #news •

Continued on next page

Table 4.7 – continued from previous page

Methods	Ranked Tweets
Amati et al. [164]	1. al Jazeera: Thousands of Egyptian youth form human shields to protect the Egyptian Museum. #Museum #Egypt #Jan25 ▶ 2 2. RT @acarvin: RT @sultanalqassemi: Al Jazeera "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" #Jan25 • 3. RT @SultanAlQassemi: Great news: Al Jazeera "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" #Jan25 • 4. RT @SultanAlQassemi: Great news: Al Jazeera "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" #Jan25 • 5. Can anyone confirm? RT @AmmarMa: Thousands of Egyptians surround the Egyptian Museum to protect it from any looting. #Jan25 • 6. RT Great news: Al Jazeera "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" • 7. Amazing// RT @niametany: Great news: Al Jazeera "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" • 8. Thousands of Egyptian youth protecting Cairo museum from sabotage. #Jan25 #Egypt #Mubarak ▶ 2 9. "Thousands of Egyptians form human-chain around Egypt Museum to protect it from looting" via AlJazeera ▶ 2 10. Great #egypt updates via Al Jazeera - follow @ajimran. So tragic. Looting the Egyptian museum and hospitals. Sad ▶ 1
Continued on next page	

Table 4.7 – continued from previous page

Methods	Ranked Tweets
Baseline	1. RT @RamyYaacoub: Confirmed: Egyptian protesters activists successfully protected the national museum from looters #Egypt #Jan25 •
	2. RT @science: Looters broke into the Egyptian Museum during anti-government protests and destroyed two Pharaonic mummies URL •
	3. RT @alihabibi1: Protesters forming teams to protect the Egyptian Museum from thieves. #Egypt #Jan25 #SidiBouzid •
	4. RT @sarahraslan: Protesters form human shield around Egyptian National Museum. Risking their lives to save their history. #Jan25 #Egypt •
	5. RT @channel4news: Egyptian army uses tanks fires shots in the air to force back hundreds of protesters attacking Central Bank building.... •
	6. RT @ianinegypt: If nobody is guarding the Egyptian Museum, who is guarding Egypt's other museums? #jan25 #egypt •
	7. al Jazeera: Thousands of Egyptian youth form human shields to protect the Egyptian Museum. #Museum #Egypt #Jan25 ▶ 2
	8. Can anyone confirm? RT @AmmarMa: Thousands of Egyptians surround the Egyptian Museum to protect it from any looting. #Jan25 •
	9. @cgorman "at National Museum... protesters formed a "human shield" around the museum to defend from possible looting..." URL ▶ 2
	10. Looters destroy mummies in Egyptian Museum URL ▶ 2

Table 4.8: Successful example of rerank the initial retrieved tweets.

Tweet ID	Current Rank	Reranked Tweets for the Query "White Stripes breakup"	Baseline Rank
32 · · · 97 ▶	1	The White Stripes announce breakup URL /say it ain't so!	1
32 · · · 04 ▶	2	I guess Jack finally realized he's a better drummer than Meg? RT Bonnaroo: What a bummer. RIP White Stripes. URL	351
32 · · · 08 ▶	3	No more White Stripes! Whatever will we do!? RT pitchforkmedia The White Stripes have officially broken up URL	6
32 · · · 36 ▶	4	White Stripes split? Does this mean Meg can concentrate on her home made porn career now? #hereshoping	329
32 · · · 12 ▶	5	It's official. Foresight doesn't make this news easier to take. RT The_AV_Club: The White Stripes break up. URL	391
32 · · · 65 ▶	6	RIP The White Stripes URL	175
32 · · · 64 ▶	7	Fell in love with The White Stripes at Glasto 2002 and seen them loads of times since. Meg will always be Jacks greatest muse. :(((	397
32 · · · 92 ▶	8	NEWS+VIDEOS — The White Stripes Call It Quits — URL (via pitchforkmedia) #Read #Breakup #RIP #Bummer	389
32 · · · 76	9	Karen Kane Women's Embroidered Peasant Top, White, Medium: Karen Kane Women's Embroidered Peasant Top, White, Me... URL	979
32 · · · 60 ▶	10	VO COM DEUS, FIAS! pitchforkmedia The White Stripes have officially broken up URL	247

Table 4.9: Unsuccessful example of rerank the initial retrieved tweets.

Tweet ID	Current Rank	Reranked Tweets for the query "British Government cuts"	Baseline Rank
34 · · · 56	1	Muslims must embrace our British values URL #ukpolitics #uknews #uk #news #politics #multiculturalism	757
34 · · · 69	2	::followergold:: UK PM blasts handling of Lockerbie case: The previous British government never exerted any pres... URL	30
34 · · · 56	3	Cameron blasts British handling of Lockerbie bomber case: The previous British government never exerted any pres... URL	3
32 · · · 60	4	@Back2LifeInc Papers: UK advised Libya on Lockerbie: British government ministers secretly advised Libya on how ... URL	21
32 · · · 88	5	UK government 'should rethink cuts and raise pension age' URL	84
34 · · · 64	6	Lockerbie bomber case blasted: The previous British government never exerted any pressure on Scottish officials ... URL	19
34 · · · 44	7	UK Government Signals May Cut Prices Paid for Renewable Energy Sources URL	96
35 · · · 84 ▶	8	GOVERNMENT spending cuts may leave the North East without Forestry Commission bases: URL #saveourforests	104
33 · · · 48	9	Budget cuts: British austerity and the price of black swan insurance – URL	48
32 · · · 72	10	British government advised the Libyan regime how to secure the release of the Lockerbie bomber. URL	24

4.4 Summary

In this chapter, we focused on the ensemble of feature sets to design an effective and efficient reranker method for microblog retrieval. We introduced two temporal features including recency and burst-aware to address the recency and temporal variation of tweets. We also alleviated the vocabulary mismatch problem by utilizing our proposed context-relevance features and three-stage query expansion technique. Along with the account related features, we introduced some popularity features to quantify the quality of a tweet. Our proposed method for estimating query type dimensions, effectively addressed the temporal and sentiment sensitivity of the query. Moreover, we applied the elastic-net regularization as a supervised feature selection technique to select the best features combination. Based on the selected features, an ensemble of learning to rank framework is used to estimate the relevance of each query-tweet pair. Experimental results on TREC Microblog 2011 and 2012 test collections over the Tweets2011 collection demonstrate the effectiveness of our method over the baseline and known related works.

However, the naive three-stage query expansion technique that we proposed in this chapter utilized the pseudo-relevant tweets at the first stage, made use of Web search results at the second stage, and extracted hashtags relevant to the query at the third stage. But for weighting terms, we only used the IDF-score of each term which might induce irrelevant rare terms from the noisy tweet contexts. Moreover, searching tweets on Twitter, users seek information with temporal relevance in mind. Therefore, highly reliant on the top retrieved results without considering temporal relevance and selecting terms by utilizing the unsupervised approach may generate the noisy or harmful expansion terms which degrade the retrieval performance [20, 21].

By considering the above limitations, in the next chapter we aim to present a query expansion framework, which augment the query by selecting the effective expansion terms under the supervised manner with the aid of ensemble of query-term relevance features.

Chapter 5

Query Expansion for Microblog Retrieval

5.1 Introduction

The rapid growth of microblog platforms such as twitter, tumblr, sina weibo, etc. provides a convenient way to the users for sharing their views, experiences, opinions, breaking news, and ideas as well as interacting with others anytime, from anywhere. Moreover, during a disaster period, such as earthquakes, floods, wildfires, and typhoons, microblogging sites are treated as an important source to serve the situational information needs [4]. That is why nowadays people are increasingly turning into microblog sites to meet their diverse information needs.

Twitter[1] has become the most popular among the microblog services. Searching tweets on Twitter, users seek information with temporal relevance in mind, such as breaking news and real-time events [8]. However, due to the length constraint of tweets, people usually use unconventional abbreviations (e.g. use "TYT" instead of "take your time," use "fab" instead of "fabulous" etc.), poor linguistic phrases (e.g. use "IMS TL;DR" instead of "I am sorry. Too long, didn't read."), and URL to express their concise thought. Besides this, some twitter specific syntaxes (e.g. #hashtags, retweets) also very popular among twitter users. All these characteristics exacerbate the severe vocabulary mismatch problem be-

[1]https://twitter.com

tween a query and a tweet which makes it challenging for effective information retrieval (IR) over tweets. For addressing the challenges of IR in microblogosphere, TREC introduced the microblog ad-hoc search task in 2011 [6]. The task is designed based on real-time information seeking behaviors, where the goal is to retrieve a set of relevant tweets based on a user's information need expressed as a query at a specific point in time.

There is a long thread of research utilizing the query expansion (QE) to mitigate the vocabulary mismatch problem in microblog retrieval [15, 16, 17, 18, 19]. Most of these methods are based on the pseudo-relevance feedback (PRF) and select the terms from the top retrieved tweets as PRF assumes the top retrieved tweets are relevant. However, highly reliant on the top retrieved results and selecting terms by utilizing the unsupervised approach may generate noisy or harmful expansion terms, which in turn degrade the retrieval performance [20, 21]. To overcome this limitation, in this chapter, we propose a query expansion method in microblog retrieval, where supervised learning is employed to select the candidate expansion terms. At first, we improve the baseline retrieval by our proposed topic modeling based query expansion technique. Next, to generate the effective source of candidate expansion terms, we introduce a convolutional long short-term memory (C-LSTM) based temporal relatedness approach along with the PRF. We consider lexical and term distribution based features, twitter specific features, temporal features, sentiment aware features, and word embedding based features to select the good expansion terms. Moreover, supervised feature selection and a state-of-the-art machine learning technique is also applied to learn the feature weight in a learning to rank (L2R) framework. Experimental results with the TREC microblog collections show that our method improves the retrieval performance over the baseline and some other competitive query expansion methods [15, 18, 19, 47, 40, 187, 188].

The main contributions of our approach are as follows: We propose a query expansion framework that augments the query by selecting the effective expansion terms under the supervised manner in microblog retrieval. To achieve this goal, we propose an effective topic modeling based query expansion technique to improve the baseline retrieval (in Section 5.2.2). We also introduce a temporal relatedness approach based on C-LSTM for candidate tweet selection which

generates the pool of effective candidate terms (in Section 5.2.3.2). To bridge the temporal and semantic gaps between the candidate terms and query, we propose new temporal, sentiment aware, and word embedding based features (in Sections 5.2.5.3, 5.2.5.4, and 5.2.5.5).

The rest of the chapter is organized as follows: **Section 5.2** provides a detailed description of our proposed query expansion approach. **Section 5.3** includes the experiments and analysis of results to show the effectiveness of our proposed method. Finally, **Section 5.4** concludes this chapter with a summary of our proposed approach.

5.2 Proposed Query Expansion Framework

Now, we describe the details of our proposed query expansion (ProposedQE) method. The goal of our query expansion technique is to alleviate the vocabulary mismatch problem by expanding the query with relevant terms, which in turn satisfy the users' information needs by retrieving more relevant tweets. We depict the overview of our proposed framework in Figure 5.1. Without the retrieval model, the rest of the parts are decomposed into Stage 1 and Stage 2, where Stage 1 is composed of Process 1 and Process 2 and Stage 2 is composed of Process 3 to Process 7.

Given a query, first we fetch the top-$\mathcal{H}$ tweets by using the baseline retrieval model. In Stage 1, we consider the top-$\mathcal{K}$ retrieved tweets to extract the expansion terms for improving the baseline retrieval through our proposed topic modeling based query expansion technique (see Fig. 1 Process 1). Then we reformulate the original query with the top-$\mathcal{M}$ expansion terms (see Fig. 1 Process 2) and fetch the top-$\mathcal{H}$ tweets again. After that, these tweets are fed into Stage 2, where we consider the top-$\mathcal{L}$ retrieved tweets and our candidate tweets selection approaches (PRF and Temporal relatedness) effectively selects two sets of candidate tweets that seem to be relevant to the query (see Fig. 1 Process 3). Once the candidate tweets are selected, we generate the pool of candidate terms through some filtering processes from each candidate tweet set (see Fig. 1 Process 4). To estimate the relevance of each term, we extract several effective features in the feature extraction stage. The features are broadly grouped into five different categories,

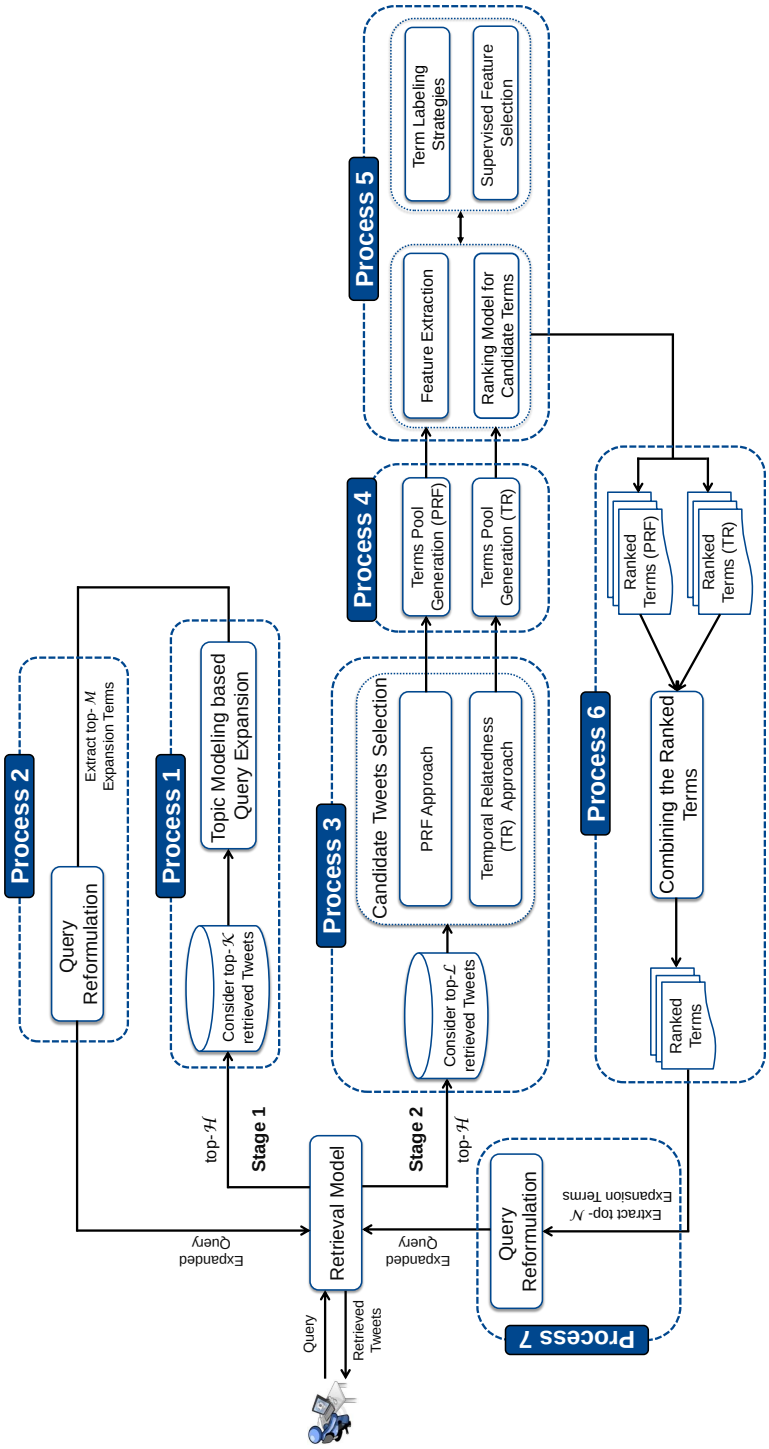

Figure 5.1: Proposed query expansion framework.

including lexical and term distribution based features, twitter specific features, temporal features, sentiment aware features, and embedding based features (see Fig. 1 Process 5). We make use of the *MinMax* [166] normalization technique for scaling the feature values. To select the best set of features, we utilize a supervised feature selection method based on *elastic-net* regularization. Next, we make use of the random forest as a feature weighting scheme to estimate the importance of the selected features (see Fig. 1 Process 5). As for ranking terms, we design a linear learning to rank (L2R) model with the aid of feature values and their importance weight (see Fig. 1 Process 5). We apply the reciprocal rank fusion technique for combining the ranked terms from PRF and temporal relatedness (TR) approaches to generate the single ranked list of terms (see Fig. 1 Process 6). Finally, we select the top-$\mathcal{N}$ expansion terms to reformulate the original query (see Fig 1 Process 7) and fetch the top-$\mathcal{H}$ tweets that are sent to the user.

5.2.1 Retrieval Model

We use the language model with Dirichlet smoothing [189] to retrieve the tweets. In the language modeling approach, each tweet in the corpus is generated by a probability distribution over the terms in the vocabulary. Given a query Q, a tweet D is ranked by the likelihood of its corresponding language model as follows:

$$f_{LM}(Q, D) = P(D|Q) \propto \ P(Q|D) \cdot P(D) \stackrel{\text{Rank}}{=} P(Q|D) \tag{5.1}$$

Assuming uniform priors over tweets and term independence:

$$P(Q|D) = \prod_{i=1}^{|Q|} P(w_i|D)$$

where $|Q|$ is the number of words in the query. Using multinomial language models, the maximum likelihood estimator of $P(w|D)$ is defined as follows:

$$P_{ml}(w|D) = \frac{n(w|D)}{|D|}$$

When the query word w does not occur in the tweet D that means, $n(w|D)$ is zero, the maximum likelihood estimate of $P(w|D)$ becomes zero and eventually

$P(Q|D)$ will be zero. To mitigate this problem, Dirichlet smoothed language model defined as follows:

$$P(w|D) = \frac{|D|}{|D| + \mu} P_{ml}(w|D) + \frac{\mu}{|D| + \mu} P(w|\mathcal{W})$$

where $P(w|\mathcal{W})$ is the collections language model and μ is the Dirichlet prior.

5.2.2 Topic Modeling based Query Expansion

Documents are modeled as a mixture of topics, where a topic is a probability distribution over words. Topics underlying within short texts (e.g. tweets) may be an important piece of information to distill its content. However, conventional topic models (e.g. latent Dirichlet allocation (LDA) [135] and probabilistic latent semantic analysis (PLSA) [136]) may not work well in this context. Due to the lack of word frequency and context information, these models suffer from the severe data sparsity problem.

We propose a topic modeling based query expansion (TMQE) which adopt the biterm topic model (BTM) [138] to tackle the data sparsity problem and generate more coherent topics from a set of tweets. The basic idea of BTM is to learn topics over tweets by directly modeling the generation of biterms in the given corpus. A biterm indicates an unordered word-pair co-occurred in a short context, where the short context refers to a proper text window containing meaningful word co-occurrences. Since top retrieved tweets are more specific to a particular query, we *locally* train the BTM model with the top retrieved tweets to perform the query-specific training in our TMQE approach.

To detect the candidate expansion terms, we propose the Algorithm 2 (TMQE), where the input is a set of top-$\mathcal{K}$ tweets, $K = \{D_1, D_2, ..., D_{\mathcal{K}}\}$ and the output is a set of candidate expansion terms, M. At first, we initialize the candidate expansion term set, M as empty. Next, we *locally* train the biterm topic model with the given set of top-$\mathcal{K}$ tweets. After that, we extract the top-$\mathcal{V}$ topics and top-$\mathcal{R}$ relevant terms of each topic. These top relevant terms are the most representative terms of a topic. However, as we train the biterm topic model with the top retrieved tweets, some terms may represent more than one topic. We hypothesize that terms which represent more than one topic are more influential

Algorithm 2: Topic Modeling based Query Expansion (TMQE) Algorithm.

Input: A set of top-$\mathcal{K}$ tweets, $K = \{D_1, D_2, ..., D_{\mathcal{K}}\}$
Output: A set of candidate expansion terms, M

1 Initialize the candidate expansion term set, $M = \varnothing$
2 Locally train the biterm topic model with a set of top-$\mathcal{K}$ tweets
3 Extract the top-$\mathcal{V}$ topics and top-$\mathcal{R}$ relevant terms of each topic
4 Generate a distinct terms pool
5 Rank the terms based on their topic coverage
6 Select the top-$\mathcal{M}$ terms to generate the candidate expansion term set, M
7 **return** M

than others. Based on this hypothesis, after generating a distinct terms pool from the extracted terms, we rank them based on their topic coverage (i.e. terms that represent the maximum number of topics get the highest rank). Finally, from this ranked list of terms, we extract the top-$\mathcal{M}$ terms to generate the candidate expansion term set, M that are linearly combined with the original query as described in Section 5.2.10.

5.2.3 Candidate Tweets Selection

After performing the topic modeling based query expansion, we utilize the expanded query to retrieve the top-$\mathcal{H}$ tweets by using the retrieval model discussed in Section 5.2.1. Our next goal is to select the tweets that will provide a source of candidate expansion terms. To get the good expansion terms, it is therefore necessary to select the more relevant tweets. In this regard, we employ two approaches: (a) Pseudo-relevance feedback (PRF) based approach and (b) Temporal relatedness (TR) approach. We utilize the top-$\mathcal{L}$ tweets from the retrieved top-$\mathcal{H}$ tweets in these approaches. Let $L = \{D_1, D_2, ..., D_{\mathcal{L}}\}$ be the set of top-$\mathcal{L}$ tweets.

5.2.3.1 Pseudo-relevance Feedback (PRF) based Approach

The PRF approach assumes that top-ranked tweets based on initial retrieval in response to the query are relevant. Because terms available in these tweets have greater probabilities to retrieve relevant tweets within that particular topic. Based on this hypothesis, we also select the top-$\mathcal{X}$ tweets from the retrieved top-$\mathcal{L}$ tweets as the candidate tweets for selecting candidate expansion terms.

5.2.3.2 Temporal Relatedness (TR) Approach

In microblog, people usually search the information about notable events or issues. An important characteristic of notable events is that they are actively discussed within a specific period of time. For example, when the breakup news of the famous band "White Stripes" was published on 2nd Feb 2011, the topic was discussed in twitter on a couple of days. After that, people lost interest in this topic and discussion of the topic was reduced. From this observation, we hypothesize that tweets that are posted in the active temporal area may contain terms relevant to the query "White Stripes breakup."

We introduce a temporal relatedness (TR) approach by exploiting the temporal distribution of retrieved top-$\mathcal{L}$ tweets to extract the candidate tweets from the active temporal area. Each tweet has an associated timestamp, which is its posting time on twitter. Let $T = \{T_1, T_2, ..., T_{\mathcal{L}}\}$ be the set of associated timestamps. Based on the associated timestamp, we cluster these tweets into different bins. Each bin corresponds to an hour-wise timestamp. Therefore, the number of bins depends on the hour-wise time span of the top-$\mathcal{L}$ tweets. Let $T_b = \{T_{b1}, T_{b2}, ..., T_{bn}\}$ be the set of temporal bins. We estimate the relatedness score of each bin by using two different approaches: (a) *Deep learning based approach* and (b) *Reciprocal rank based approach.*

In the first approach, we utilize a popular deep learning technique to estimate the relatedness score of each temporal bin, whereas in the second approach we utilize the rank of the tweets to estimate the bin score. Finally, we combine them both to estimate the final relevance score of each bin.

We choose the deep learning methods in the first approach because traditional bag-of-words based methods cannot perform well due to the curse of dimensional-

ity and the loss of word order information. However, to estimate the relatedness score using deep learning methods, it is required to represent tweets as meaningful features. Therefore, for effective tweet representation, we utilize the C-LSTM architecture. We consider each temporal bin as a category and train the C-LSTM model to determine the temporal bins that contain the most relevant tweets to the query.

While several variants of C-LSTM exist, we utilize the architecture proposed by Zhou et al. [65] for our purpose. In this architecture, the higher level representations of CNN are fed into the LSTM to learn long-term dependencies. The CNN is constructed on top of the pre-trained word vectors from large tweet corpus to learn higher-level representations of n-grams. The feature maps of CNN are then organized as sequential window features to serve as the input of LSTM to learn sequential correlations from higher-level sequence representations. The LSTM transition functions are defined as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
u_t &= \phi(W_u \cdot [h_{t-1}, x_t] + b_u) \\
c_t &= f_t \odot c_{t-1} + i_t \odot u_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where i_t, f_t, o_t, u_t, c_t, and h_t denote the input gate, forget gate, output gate, cell input activation, the cell state, and the current hidden state, respectively, at the current time step t. The symbol σ is the logistic sigmoid function to set the gating values in $[0, 1]$. ϕ is the hyperbolic tangent activation function that has an output in $[1, -1]$ and $\odot$ is the element-wise multiplication.

At the last time step of LSTM, the output of the hidden state is regarded as the tweet representation and passed to a fully connected softmax layer on top. The output of the softmax layer is the probability distribution over all the categories. We consider cross-entropy as the loss function and train the model by minimizing the error, which is defined as:

$$
E(x^{(i)}, y^{(i)}) = \sum_{j=1}^{k} 1\{y^{(i)} = j\} \log(\tilde{y_j}^{(i)})
$$

where $x^{(i)}$ is the training sample with its true label $y^{(i)}$. $\tilde{y_j}^{(i)}$ is the estimated probability in $[0, 1]$ for each label j. $1\{condition\}$ is an indicator which is 1 if true and 0 otherwise. To learn the model parameter, we utilize the stochastic gradient descent (SGD) and adopt the Adam optimizer [190].

After completion of the model training, we pass the original query to that model and get the relevance score of all the temporal bins. We renamed this relevance score as semantic similarity (*SemSim*) score.

In the second approach, we utilize the rank of each tweet based on the initial retrieved result to estimate the score of each bin. We hypothesize that the bin which contains more top-ranked tweets will be related to the query.

We combine both approaches to compute the final relevance score of each temporal bin as follows:

$$\mathcal{RST}(T_{bi}) = SemSim(Q, T_{bi}) + \sum_{D_j \in T_{bi}} \frac{1}{rank_{D_j}}$$

where $rank_{D_j}$ is the rank of the tweet in the initial retrieved result, the first component defines the semantic similarity based on C-LSTM, and the second component estimates the bin's score based on tweets rank.

Finally, to select the candidate tweets, we rank all the temporal bins based on the $\mathcal{RST}$ score and select the top-$\mathcal{Y}$ bins. Then, we take the top-ranked tweets (based on the initial retrieved result) from each of the selected bins to construct the candidate tweet set.

5.2.4 Terms Pool Generation

Once the candidate tweets are selected by using the approaches described in Section 5.2.3, we generate the pool of candidate terms for each candidate tweet set. In this regard, we tokenize the tweets and remove all the available stopwords. Finally, the pool contains all the unique terms available in the selected tweets. In order to select the effective expansion terms, we need to rank these terms. To achieve this goal, we extract several term relevance features that are presented next.

5.2.5 Feature Extraction for Candidate Terms

After generating the candidate terms pool from each of the two candidate tweet sets, we extract a set of 25 features for selecting a good set of candidate expansion terms. These features are grouped into 5 different categories. Table 5.1 presents all these features for our term learning to rank framework. We utilize the top-$\mathcal{L}$ retrieved tweets to estimate the embedding based features and the first 3 temporal features, whereas to estimate the rest of the features, we only utilize the selected candidate tweets. Let $CT = \{C_1, C_2, ..., C_m\}$ be the set of candidate terms and $DT = \{D_1, D_2, ..., D_n\}$ be the set of candidate tweets. Hence, the feature extraction processes of each candidate term C are described next.

5.2.5.1 Lexical and Term Distribution based Features

We extract 12 lexical and term distribution based features, where the first six features are document frequency (DF) [31], inverse document frequency (IDF) [31], TF-IDF [31], inverse corpus frequency (ICF) [192], linearly discounted IDF [191], and Okapi BM25 [34]. We also extract the co-occurrence based term weighting features including co-occurrence with single query term and co-occurrence with the pair of query terms, proposed by Cao et al. [20].

However, there is a chance that terms frequently co-occur with most of the query terms tend to discriminate poorly between the relevant and non-relevant documents or tweets [193]. Considering this fact, we devise two co-occurrence based features which favor the candidate term co-occur with the minimum number of query terms. Therefore, given a query Q, the score of the candidate term, C is estimated with the aid of the inverse exponential function of the number of query terms occur in the tweet, where the candidate term also occurs. The weight is the sum of its scores over all the candidate tweets as follows:

$$f_{TCF}(C, Q, DT) = \sum_{D \in DT : C \in D} e^{-MQT}$$

where $MQT = |q_i \in Q : q_i \in D \cap C \in D|$ is the number of query terms available in the tweet D which also contains the candidate term, C and $MQT \neq 0$. Along with this direction, we also utilize a variant of this feature as follows:

$$f_{TCFV}(C, Q, DT) = e^{-\sum_{D \in DT : C \in D} MQT}$$

Table 5.1: List of features, where our proposed features are highlighted in bold.

Feature Type	Feature Name
Lexical and Term Distribution based Features	1. Document Frequency (DF) [31] 2. Inverse Document Frequency (IDF) [31] 3. Linearly Discounted IDF [191] 4. TF-IDF [31] 5. Inverse Corpus Frequency (ICF) [192] 6. Okapi BM25 [34] 7. Co-occurrence with single query term [20] 8. Co-occurrence with the pair of query terms [20] 9. **Co-occur with the minimum number of query terms (TCF)** 10. **Variants of 9 no. feature (TCFV)** 11. **Average Tweet Length (ATL) Feature** 12. **Parts-of-Speech (POS) Feature**
Twitter Specific Features	1. **Hashtag (HT) Feature** 2. **Hashtag Popularity (HTP) Feature**
Temporal Features	1. **Maximum Time Series Similarity** 2. **Minimum Time Series Similarity** 3. **Mean Time Series Similarity** 4. **Temporal Distance Feature** 5. **Minimum Temporal Distance Feature**
Sentiment Aware Features	1. **Sentiment Polarity (SP) Feature** 2. **Sentiment Match (SM) Feature**
Embedding based Features	1. **Mean Cosine Similarity** 2. **Maximum Cosine Similarity** 3. **Minimum Cosine Similarity** 4. **Linearly Discounted Score**
Total	25 Features

Average Tweet Length (ATL) Feature: Intuitively, a longer tweet may be able to carry more information. From this intuition, we extract the average tweet length feature for a candidate term. To estimate this feature, we consider those tweets in the candidate tweet set, DT that contains the candidate term, C and estimate this feature as follows:

$$f_{ATL}(C, DT) = \frac{1}{|D \in DT : C \in D|} \sum_{D \in DT : C \in D} len(D)$$

where $|D \in DT : C \in D|$ denotes the number of tweets that contain the candidate term, C and $len(D)$ is the tweet length. We estimate the tweet length by counting the number of terms it contains.

Parts-of-Speech (POS) Feature: To estimate the informativeness of a candidate term, we intend to distill its POS information. We hypothesize that a candidate term is important if its POS is either a noun or adjective. Our POS feature is a binary feature that is assigned to 1 if the candidate term's POS is either a noun or adjective and 0 otherwise.

$$f_{POS}(\mathcal{C}) = \begin{cases} 1, \text{ if POS}(\mathcal{C}) \in PL \\ 0, \text{ otherwise} \end{cases}$$

where $PL = \{\text{Noun, Adjective}\}$

5.2.5.2 Twitter Specific Features

Twitter has some special characteristics. One of them is a hashtag, which is very popular among the users. A twitter hashtag is a type of metadata tag that highlights the important topic or event on twitter. We extract two features based on the hashtag to rank the candidate terms.

Hashtag (HT) Feature: Since a hashtag highlights the important topic or event on twitter, the candidate term which is used as the hashtag might have some importance. Based on this intuition, we define our binary hashtag feature that is assigned to 1 if the candidate term is a #Hashtag and 0 otherwise.

$$f_{HT}(C) = \begin{cases} 1, \text{ if the candidate term } C \text{ is a \#Hashtag} \\ 0, \text{ otherwise} \end{cases}$$

Hashtag Popularity (HTP) Feature: Besides the binary hashtag feature, we also extract a feature based on hashtag popularity. For a given candidate term C, the hashtag popularity feature is estimated as follows:

$$f_{HTP}(C, DT) = e^{\sum_{\#Ht \in DT:C \in \#Ht} TF(\#Ht)}$$

where $TF(\#Ht)$ is the frequency of the hashtag among the candidate tweets DT that contains the candidate term.

5.2.5.3 Temporal Features

Since users usually search for real-time news and events information in microblog, temporal information needs to be considered for effective information retrieval in the microblog. To extract the temporal aspect of candidate terms, we extract our proposed temporal features such as time series similarity and temporal distance by leveraging the temporal correlation of candidate term and query terms.

Maximum Time Series Similarity: To estimate the maximum time series similarity (MaxTSS) feature, we apply a similar kind of temporal binning process described in Section 5.2.3.2. However, instead of hour-wise temporal binning, we cluster the top-$\mathcal{L}$ retrieved tweets into 15min-wise temporal bins. We hypothesize that terms that occur in similar temporal bins would be relevant to each other. In this regard, we utilize the word-occurrence distribution of each of the candidate and query terms in the temporal bins to approximate its time series. Then, we estimate the similarity measure of two time series generated for two terms to quantify the temporal relatedness between them.

To illustrate this hypothesis, a sample of time series curves from two instances "BBC" and "Language" are shown in Figure 5.2. Here, X-axis denotes the temporal bins and Y-axis denotes the frequency of these terms labeled as word intensity in the respective temporal bins. We can see that both curves have fluctuated in the similar temporal regions, which in turn deduce their relatedness.

On the basis of the time series of candidate term and each of the query term, we estimate the maximum time series similarity (MaxTSS) feature as follows:

$$f_{MaxTSS}(C, Q) = \max_{q_i \in Q} dCor\ (T_C, T_{q_i})$$

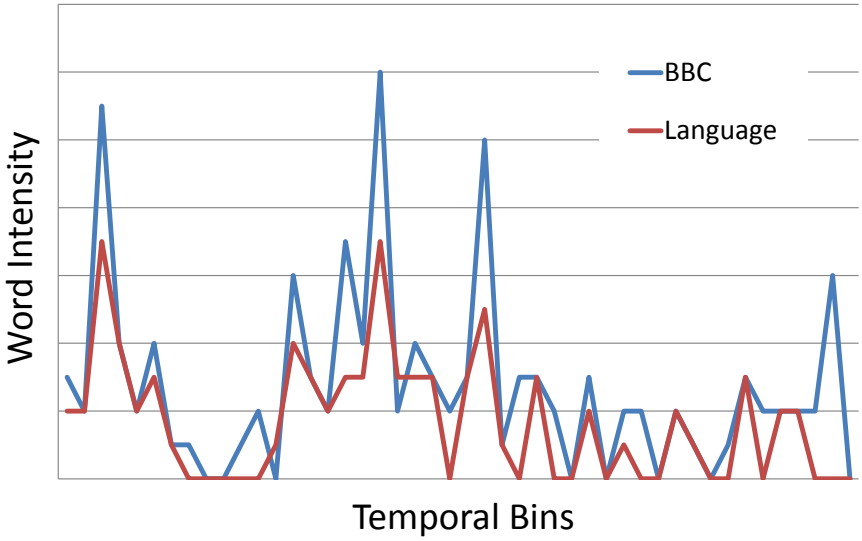

Figure 5.2: Time series representation of two sample terms.

where T_C denotes the time series of the candidate term C, T_{q_i} denotes the time series of the query term q_i, and $dCor$ is the distance correlation function [194]. The $dCor$ of T_C and T_{q_i} is obtained by dividing their distance covariance ($dCov$) by the product of their distance standard deviations as follows:

$$dCor(T_C, T_{q_i}) = \frac{dCov\ (T_C, T_{q_i})}{\sqrt{dVar\ (T_C)\ \ dVar\ (T_{q_i})}}$$

where $dVar$ is the distance variance. If the $dCor$ value is closed to 1, it indicates that two time series are nearly similar. Whereas they are dissimilar if the $dCor$ value is closed to 0.

Along with this direction, we also proposed the following two features named as minimum time series similarity (MinTSS) and mean time series similarity (MeanTSS), which are the variants of maximum time series similarity (MaxTSS) feature.

Minimum Time Series Similarity: We estimate the minimum time series similarity (MinTSS) as a feature based on the time series of candidate term and query terms as follows:

$$f_{MinTSS}(C, Q) = \min_{q_i \in Q} dCor\ (T_C, T_{q_i})$$

Mean Time Series Similarity: We also estimate the mean time series similarity (MeanTSS) as a feature based on the time series of candidate term and query terms as follows:

$$f_{MeanTSS}(C, Q) = \frac{1}{|Q|} \sum_{q_i \in Q} dCor\ (T_C, T_{q_i})$$

Temporal Distance Feature: We hypothesize that if the candidate term and query are temporally close, it is more likely that they are relevant. Based on this hypothesis, we define a temporal distance feature based on the temporal vicinity of the query time and the candidate tweet time having candidate term. The weight of the term is obtained by summing its scores over all the candidate tweets with the aid of exponential score function. We also utilize the rank of the candidate tweet and IDF score of the term. Therefore, we estimate the temporal distance feature as follows:

$$f_{TD}(C, Q, DT) = \sum_{D \in DT : C \in D} \frac{1}{rank_D} \cdot \delta e^{-\delta * \log |Q_{Time} - D_{Time}|} \cdot\ IDF\ (C)$$

where Q_{Time} is the query time, D_{Time} is the candidate tweet time that contains the candidate term C, $rank_D$ is the rank of the candidate tweet D, and $IDF\,(C)$ is the inverse document frequency of the candidate term C in the candidate tweets. The rate parameter, δ is set to 0.001.

Minimum Temporal Distance Feature: Instead of summing score over all the candidate tweets, we also estimate the minimum temporal difference between the query time and the candidate tweet time having the candidate term. Therefore, we estimate the feature as follows:

$$f_{MTD}(C, Q, DT) = \frac{1}{rank_D} \cdot \delta e^{-\delta * MinTD} \cdot\ IDF\ (C)$$

where $IDF\,(C)$ is the inverse document frequency of the candidate term C among the candidate tweets, $rank_D$ is the rank of the candidate tweet D, which has the minimum time difference with respect to the query time and

$$MinTD = \min_{D \in DT : C \in D} \Big(\log |Q_{Time} - D_{Time}| \Big)$$

5.2.5.4 Sentiment Aware Features

As notable events are usually sentiment sensitive and tweets reflect people's opinions and attitudes, therefore it is important to extract the sentiment aspect of candidate terms. In this regard; we introduce two sentiment aware features by utilizing the sentiment of the candidate term and query.

Sentiment Polarity (SP) Feature: To reward the sentimentally sensitive candidate terms, we propose a binary sentiment polarity (SP) feature that is assigned to 1 if the candidate term, C has the positive or negative sentiment polarity and 0 otherwise.

$$f_{SP}(C) = \begin{cases} 1, \text{ if } C \text{ has the sentiment polarity} \\ 0, \text{ otherwise} \end{cases}$$

Sentiment Match (SM) Feature: To reward the candidate terms that are sentimentally identical to the query, we propose a sentiment match (SM) feature based on the query sentiment and candidate term's sentiment. Our sentiment match feature is a binary feature that is assigned to 1 if the sentiment of the candidate term and query are identical and 0 otherwise.

$$f_{SM}(C, Q) = \begin{cases} 1, \text{ if } C_s = Q_s \\ 0, \text{ otherwise} \end{cases}$$

where C_s and Q_s denote the sentiment polarity of the candidate term and query, respectively.

5.2.5.5 Embedding based Features

A word embedding is a mapping that associates words occurring in a collection to a vector in R^n, where n is significantly lower than the size of the vocabulary of the collection [38]. If we consider two words A and B, then the distance between these words in the embedding space indicate a quantitative semantic relatedness between them. Therefore, to estimate the semantic relatedness of a candidate expansion term with the query terms, we introduce four features by leveraging word embedding. We *locally* train the *word2vec*[1] model proposed by Mikolov et al. [39] with the top-$\mathcal{L}$ retrieved tweets to get the word vector representation.

[1] *word2vec* (https://code.google.com/p/word2vec/)

Mean Cosine Similarity: We estimate the mean cosine similarity (MeanCosSim) score based on the word vector representation of the candidate term, C and all the query terms in the embedding space, as follows:

$$f_{MeanCosSim}(C, Q) = \frac{1}{|Q|} \sum_{q_i \in Q} \vec{C} \cdot \vec{q_i}$$

Maximum Cosine Similarity: We estimate the maximum cosine similarity (MaxCosSim) score based on the word vector representation of the candidate term, C and all the query terms in the embedding space, as follows:

$$f_{MaxCosSim}(C, Q) = \frac{1}{|Q|} \max_{q_i \in Q} (\vec{C} \cdot \vec{q_i})$$

Minimum Cosine Similarity: We also estimate the minimum cosine similarity (MinCosSim) score based on the word vector representation of the candidate term, C and all the query terms in the embedding space, as follows:

$$f_{MinCosSim}(C, Q) = \frac{1}{|Q|} \min_{q_i \in Q} (\vec{C} \cdot \vec{q_i})$$

Linearly Discounted Score: Instead of using the similarity score between the candidate term and query terms, here we utilize the rank position of the candidate term in each of the query term's embedding space. In this regard, we extract the top-ranked 1000 most similar words based on cosine similarity for each of the query terms by utilizing the *locally* trained *word2vec* model. We then estimate the linearly discounted score between the candidate term, C and all the query terms as follows:

$$f_{LDS}(C, Q) = \frac{1}{|Q|} \sum_{q_i \in Q} f_{rank}(C, TW_{q_i})$$

where

$$f_{rank}(C, TW_{q_i}) = \begin{cases} \frac{1}{rank_C}, & \text{if } C \in TW_{q_i} \\ 0, & \text{otherwise} \end{cases}$$

where TW_{q_i} is the set of top 1000 most similar words of the query term q_i and $rank_C$ is the rank of the candidate term that appears in the TW_{q_i}. The value of the feature function, f_{rank} will be increasingly reduced if the candidate term C appears in the lower rank of the most similar word set TW_{q_i} and is set to 0 otherwise.

5.2.6 Term Labeling Strategies

To rank the candidate terms in a supervised manner, it is necessary to assign the label of the terms such as relevant or non-relevant. Intuitively, given a query, a good expansion term improves the retrieval performance while combining with the original query terms, whereas a bad expansion term hurts the performance [21].

Therefore, to estimate the label of each candidate term, we consider the retrieval performance by adding this term to the original query. If the retrieval performance increases, that means the term has a positive impact on retrieval (i.e. relevant term) and we assign 1 to it as its label and 0 otherwise. Suppose $eval(Q)$ and $eval(Q \cup C)$ are the performance evaluation of the original query and expanded query with the candidate term C, respectively. We estimate the performance difference due to the expansion term, C and the relevance label (RL) for the candidate term is assigned as follows:

$$RL(C) = \begin{cases} 1, & \text{if } \mathit{Diff}\,(C) > 0 \\ 0, & \text{otherwise} \end{cases}$$

where

$$\mathit{Diff}\,(C) = eval\ (Q \cup C) - eval\ (Q)$$

We use the precision@30 (P@30) as the evaluation measure to estimate the $eval\ (Q \cup C)$ and $eval\ (Q)$.

5.2.7 Supervised Feature Selection

Feature selection is the process of selecting the most relevant features to enhance the performance of the predictive model. To improve the performance of our candidate term selection approach in our proposed query expansion framework, we employ the elastic-net regularized regression [175, 47] as a supervised feature selection (SFS) method that selects the best set of features through eliminating the irrelevant features.

5.2.8 Ranking Model for Candidate Terms

We employ a linear learning to rank (L2R) model to estimate the relevance score of each candidate term. For a given query Q and a candidate term C, the term

relevance score value, *rsv* is estimated as follows:

$$rsv(Q, C) = \frac{\sum_{i=1}^{N_f} \lambda_i f_i(Q, C)}{\sum_{i=1}^{N_f} \lambda_i} \tag{5.2}$$

where $f_i(Q, C)$ is the feature function, N_f is the number of features, and λ_i is the model parameter. To set this model parameter λ_i, we make use of one state-of-the-art machine learning model named as random forest. We estimate the *MeanDecreaseGini*, a measure of variable importance in random forest model. The *Gini* impurity value for the two descendant nodes is less than the parent node every time a split of a node occurred on a certain feature f. The importance score of each feature is estimated by summing up the *Gini* decreases for each individual feature over all trees in the forest [180]. We use this importance score of each feature to set the model parameter, λ_i.

5.2.9 Combining the Ranked Terms

Once we get the ranked candidate terms that are generated from PRF approach and temporal relatedness (TR) approach based candidate tweet sets, we adopt the reciprocal rank fusion (RRF) method [177] in our system. The RRF method combines these two rankings of candidate terms to generate a single rank list of terms, with the aim of improving over the performance of individual ranking. To achieve this, RRF sorts the candidate terms based on a naive scoring function. For a given set of candidate terms, CT to be ranked and a set of term rankings $\mathcal{R}_T$, each a permutation on $1 \cdots |CT|$, RRF_{score} of each candidate term is estimated as follows:

$$RRF_{score}(C \in CT) = \sum_{r \in \mathcal{R}_T} \frac{1}{\gamma + r(C)} \tag{5.3}$$

where $r(C)$ is the rank of the candidate term, C and the constant γ is used to alleviate the impact of high rankings by outlier systems.

Based on the RRF_{score}, we sort the candidate expansion terms and get the final rank list. Finally, from this ranked list, we extract the top-$\mathcal{N}$ terms as candidate expansion terms that are linearly combined with the original query terms as described in the Section 5.2.10.

5.2.10 Query Reformulation

Once the expansion terms are selected, we estimate the expanded query likelihood model $P(Q_{Expanded}|D)$ as a linear combination of the two respective query likelihoods, one for the original query and the other for the expansion terms. Therefore, the expanded query likelihood model is obtained by:

$$P(Q_{Expanded}|D) = \alpha \cdot P(Q_{Original}|D) + (1 - \alpha) \cdot P(Q_{ExpTerms}|D) \tag{5.4}$$

where $P(Q_{Original}|D)$ is the query likelihood model based on the original query, $P(Q_{ExpTerms}|D)$ is the query likelihood model based on the expansion terms, and α is the anchoring weight parameter. The expanded query likelihood model $P(Q_{Expanded}|D)$ is then used instead of $P(Q|D)$ in Eq. (5.1) in Section 5.2.1 to retrieve the tweets.

5.3 Experiments and Evaluation

5.3.1 Experimental Setup

Dataset Collection: In order to assess our proposed query expansion (QE) method, we made use of publicly available Tweets2011[1] corpus used in the TREC Microblog 2011 (TMB2011) [6] and 2012 (TMB2012) [114] tracks. The collection consisted of approximately 16 million tweets sampled from twitter over a period spanning from January 23, 2011 to February 7, 2011 (inclusive). Popular events that happened during this period include democracy revolution in Egypt, US superbowl, BBC service cut, and so on. As Twitter's terms of service forbid the redistribution of tweets, TREC organizers provided a streaming API to crawl the corpus. Using this official TREC Microblog API [6], we generated our local Tweets2011 corpus. There are 50 timestamped topics released for the TMB2011 track and 60 timestamped topics released for the TMB2012 track. The associated relevance judgments of tweets provided by the organizers for these query topics consisted of three relevance levels, including irrelevant (labeled 0), relevant (labeled 1), and highly relevant (labeled 2). In TMB2011 query set; 49 queries

[1] http://trec.nist.gov/data/tweets/

<top>
<num> Number: MB002 </num>
<title> 2022 FIFA soccer </title>
<querytime> Tue Feb 08 18:51:44 +0000 2011 </querytime>
<querytweettime> 35048150574039040 </querytweettime>
</top>

Figure 5.3: Query sample.

have at least one relevant or highly relevant tweet and 33 queries have at least one highly relevant tweet. On the other hand, in TMB2012 query set; 59 queries have at least one relevant or highly relevant tweet and 56 queries have at least one highly relevant tweet. We evaluated our proposed method in descending order of relevance for both *allrel* and *highrel* relevance criteria. In *Allrel* relevance criteria, both relevant and highly relevant tweets are considered as relevant, whereas only highly relevant tweets are considered as relevant in *highrel* relevance criteria. The basic statistics of the query sets and relevance judgments shown in Table 5.2.

Table 5.2: The statistics of TMB2011-12 query sets and relevance judgments.

Category	TMB2011	TMB2012
Number of Topics	50	60
Number of Annotated Tweets	40855	73073
Number of Annotated Nonrel Tweets	37991	66787
Number of Annotated Allrel Tweets	2864	6286
Number of Annotated Highrel Tweets	558	2572

Each tweet document is composed of tweet_id, tweet_time, and tweet_text; whereas each query topic is composed of query_number, query_text, query_time, and query_tweet_time. An example query is shown in Figure 5.3. The query number is enclosed by the *num* tag; whereas the query text is enclosed by the *title* tag. The *querytime* tag defines the timestamp of the query in ISO format. The *querytweettime* defines the ID of the tweet which timestamp defines the query timestamp. Therefore, tweets whose IDs' are not greater than this ID need to be considered for this query.

Data Preprocessing: The preprocessing step is initiated with a filtering process that refines the crawled corpus based on non-English tweet removal, retweet removal, and future tweet removal. Even though twitter is a multilingual microblog service, only English tweets were judged as relevant in this research. Therefore, a language detection library[1] was applied to remove the non-English tweet from the corpus. In addition, retweets were eliminated from the corpus as they are just the identical copy of other tweets and did not provide any additional information. Tweets that start with the word "RT" were identified as retweets. Moreover, tweets often contain non-standard word forms and domain-specific entities. For example, people usually use "Birrrtthhdaayy" instead of "Birthday," "celebs" instead of "celebrities," "mktg" instead of "marketing," etc. We utilized two lexical normalization dictionaries collected from [167] and [168] to normalize such non-standard words into their canonical forms. Also all non-English characters were removed from the tweets. Along with this direction, future tweets were discarded from the corpus for the individual query. All tweets that are posted after the timestamp of the query were regarded as future tweets. Throughout the experiments, we did not remove stopword due to the brevity of tweets except candidate terms pool generation. We applied the Indri's standard stoplist[2] for stopword removal.

Feature Importance Estimation: We employed a publicly available package glmnet [178] for supervised feature selection using the elastic-net regularization method. The result of our supervised feature selection process showed that *Hashtag (HT)* and *inverse corpus frequency (ICF)* features are irrelevant.

Next, we made use of a publicly available package of random forest [179] to estimate the importance of selected features. We utilized this package to estimate the *MeanDecreaseGini* [180], a measure of variable importance in the random forest model. A ranked list of our selected features based on the importance score is illustrated in Figure 5.4, where our proposed features are boldfaced.

Among all the 23 selected features, our proposed W2V Maximum Cosine Similarity feature and Mean Time Series Similarity feature ranked at position first and

[1]https://code.google.com/p/language-detection/

[2]https://www.lemurproject.org/stopwords/stoplist.dft

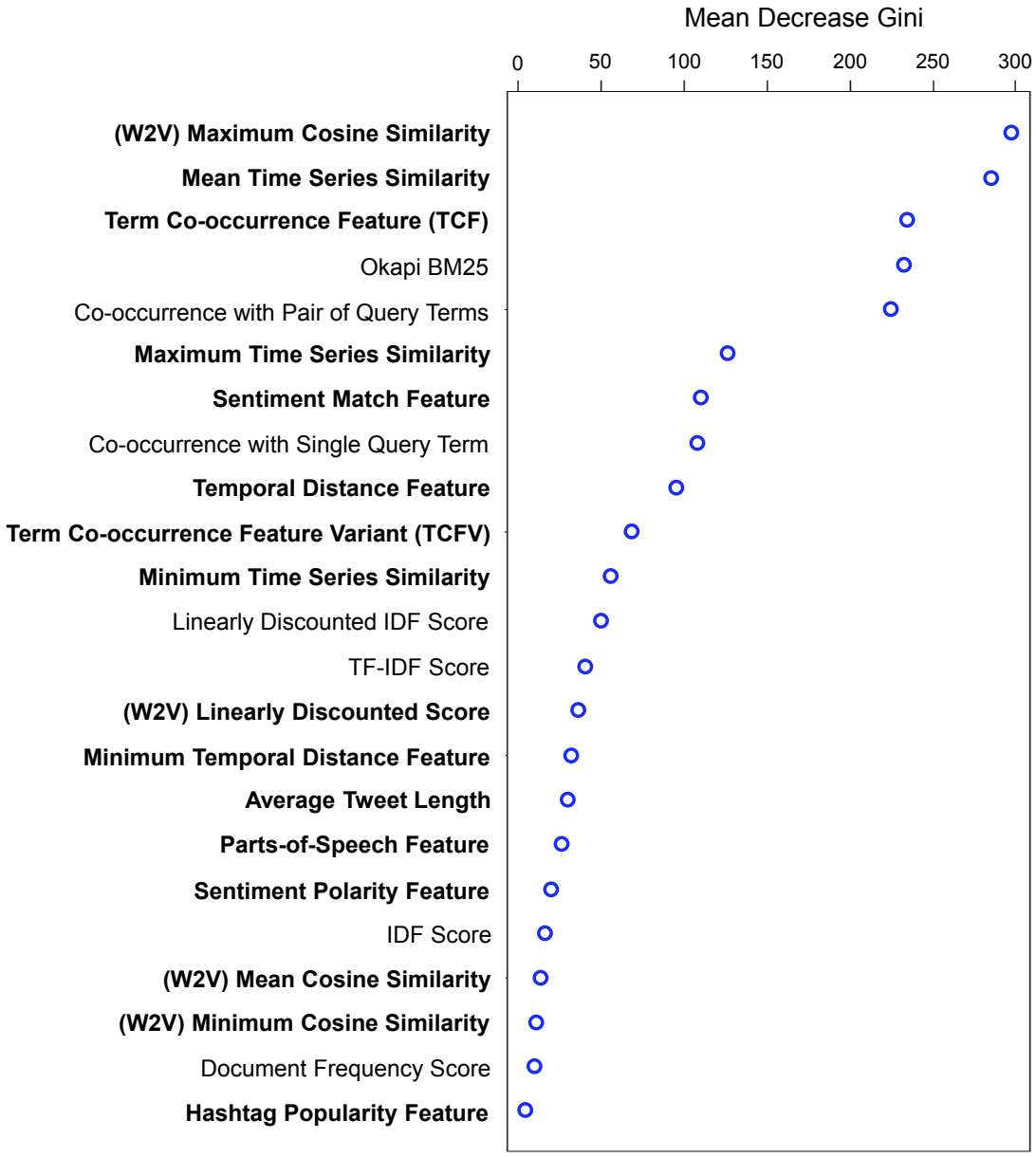

Figure 5.4: Feature importance.

second, which demonstrated the complementary importance of embedding based feature and temporal feature. However, other proposed temporal features were ranked at the sixth, ninth, eleventh, and fifteenth position, whereas other variants of embedding based features were ranked at position fourteenth, twentieth, and twenty-first position, respectively. Along with this direction, our proposed

sentiment features were ranked at the seventh and eighteenth position, which in turn deduce the importance of considering sentiments for selecting good expansion terms. The rest of the proposed features including term co-occurrence feature (TCF) and its variant TCFV feature, average tweet length feature, POS feature, and hashtag popularity feature were ranked at the third, tenth, sixteenth, seventeenth, and twenty-third position, respectively. From this observation, we can deduce that our proposed features are effective for selecting good expansion terms and can enhance the performance of query expansion technique in microblog search.

Training and Testing L2R Model: For candidate term ranking, we applied a linear learning to rank (L2R) model as stated in Eq. (5.2). In this regard, we made use of a publicly available package of random forest [179] with no parameter tuning. To set the model parameter λ_i in Eq. (5.2), we utilized the feature importance scores (*MeanDecreaseGini* [180]) of our selected features obtained from the random forest. In our settings, at first we trained on TMB2011 queries to learn the feature importance score and test on TMB2012 queries, and vice versa.

Parameters Setting: In the following, we describe the set of parameters that we have used in our experiments. We utilized the Lucene[1] framework to index our corpus and used the Lucene's implementation of query-likelihood (LMDirichletSimilarity) as our retrieval model. In all of our experiments, the Dirichlet prior smoothing parameter μ was set to 2000 (see Section 5.2.1) and we retrieved the top-$\mathcal{H} = 1000$ tweets.

For our topic modeling based query expansion (TMQE) approach, we utilized the top-$\mathcal{K} = 50$ tweets retrieved by the retrieval model to *locally* train the BTM model. We empirically extracted the top-$\mathcal{V} = 10$ topics and top-$\mathcal{R} = 10$ relevant terms of each topic (see Algorithm 2). To select the optimal number of feedback terms, we performed the grid search and the optimal number of feedback terms was set as top-$\mathcal{M} = 3$ for both the TMB2011 and TMB2012 test set, respectively.

[1]https://lucene.apache.org/core/

However, for the candidate tweet selection approach including PRF and temporal relatedness (TR) based approach, we utilized the top-$\mathcal{L}$ fetched tweets to extract the top-$\mathcal{X}$ and top-$\mathcal{Y}$ candidate tweets, respectively. We set top-$\mathcal{L} = 1000$ and the number of candidate tweet was set to the typical value of 30 (i.e. top-$\mathcal{X} = 30$, and top-$\mathcal{Y} = 30$). Because Miyanishi et al. [19] reported that the performance is not sensitive when the number of candidate tweet is sufficiently large (i.e. $\geqslant 30$).

The C-LSTM model used in our temporal relatedness (TR) approach for candidate tweet selection was based on Theano [195] and trained on a GPU to capture the benefit from the efficiency of parallel computation of tensors. We performed hyper-parameter optimization using a simple grid search. Our final C-LSTM model contains one convolutional layer and one LSTM layer. We utilized a *word2vec* model pre-trained on Tweets2011 corpus. We embedded the *word2vec* model in a 300-dimensional space and used the skip-gram model with negative sampling. The number of negative examples was set to 5, the width of the word-context window i.e. window size was set to 8, and we discarded the words that appear fewer than 2 times i.e. min_count was set to 2. Both the CNN layer and the LSTM layer were dropped out with a probability of 0.2. L2 regularization with a factor of 0.0001 was applied to the weights in the softmax layer. During the training, the number of class label was equal to the number of temporal bins of each query.

Like the candidate tweet selection approach, we also utilized the top-$\mathcal{L} = 1000$ retrieved tweets to estimate the embedding based features and the time series based features. However, only selected candidate tweets (i.e. top-$\mathcal{X} = 30$ and top-$\mathcal{Y} = 30$) were used for other features. As already mentioned earlier, during the estimation of embedding based features we *locally* trained the *word2vec* model. We used the similar parameter settings that we used in our C-LSTM model to *locally* train the *word2vec* model except for the window size, which was set to 5 here. Moreover, to compensate for the smaller corpus in the local *word2vec* model, we performed the 20 iterations during the model training. To identify the POS of each candidate expansion terms during the estimation of POS feature, we utilized the CMU ARK POS tagger [196]. Along with this direction, a publicly available package SentiStrength [181] was applied to estimate the sentiment of candidate

term and query during the estimation of sentiment aware features. According to the recommendation by Cormack et al. [177], we set the constant, γ in Eq. (5.3) as 60.

To select the optimal number of feedback terms, top-$\mathcal{N}$ in our proposed query expansion (QE) method, we performed the grid search based on both the TMB2011 and TMB2012 test collections. The optimal number of feedback terms was selected from {5,10,.....,50}. Another parameter in our method is the anchoring parameter, α, that we used to combine the query likelihood models obtained from the expansion terms and original query terms as shown in Eq. (5.4). To select the optimal value, we swept the parameter between $\{0.1, ..., 0.9\}$. Unless otherwise stated, default settings were used for the other parameters.

5.3.2 Results with Query Expansion

We now evaluate the retrieval effectiveness of our proposed query expansion method. In this regard, we employed four evaluation measures, including precision at top 30 tweets (P@30), mean average precision (MAP), normalized discounted cumulative gain at top 30 tweets (NDCG@30), and R-Precision (R-Prec). A detailed description of these evaluation measures was disucussed in the chapter 4 on Section 4.3.2. Following the TREC microblog benchmark [6], we also considered P@30 as the primary evaluation measure. We used a two-sided paired t-test at 95% confidence level for statistical significance testing between two systems' performances, where † denotes the statistically significant at ($p < 0.05$). The summarized results of our experiments were presented in Table 5.3 and Table 5.4, respectively.

At first, we showed the retrieval performance based on baseline, which is Lucene's implementation of query-likelihood (LMDirichletSimilarity) model. Results based on two different query expansion approaches were presented in the *TMQE* and *ProposedQE* setting, respectively. In the *TMQE* setting, the topic modeling based query expansion approach (discussed in Section 5.2.2) was used to expand the query. The optimal number of feedback terms, top-$\mathcal{M}$ was set to 3 and the anchoring parameter, α was set to 0.6 by using grid search. Whereas, in the *ProposedQE* setting, our proposed query expansion (QE) approach was

Table 5.3: Performance (P@30, R-Prec, MAP, and NDCG@30; higher is better) on TMB2011 test set for various experimental settings. The best results are highlighted in boldface. † indicates the statistically significant difference between the baseline and each method at ($p < 0.05$).

Method	Allrel				Highrel		
	P@30	R-Prec	MAP	NDCG@30	P@30	R-Prec	MAP
Baseline	0.3483	0.3509	0.3050	0.4374	0.1253	0.2405	0.2378
TMQE	0.4537†	0.4209†	0.3921†	0.5147†	0.1707†	0.2389	0.2373
ProposedQE	**0.5041**†	**0.4301**†	**0.4222**†	**0.5518**†	**0.1798**†	**0.2454**	**0.2554**

Table 5.4: Performance (P@30, R-Prec, MAP, and NDCG@30; higher is better) on TMB2012 test set for various experimental settings. Legend settings are identical to Table 5.3.

Method	Allrel				Highrel		
	P@30	R-Prec	MAP	NDCG@30	P@30	R-Prec	MAP
Baseline	0.2932	0.2354	0.1815	0.2862	0.1625	0.1845	0.1388
TMQE	0.4034†	0.3379†	0.2742†	0.3799†	0.2167†	0.2182†	0.1927†
ProposedQE	**0.4723**†	**0.3615**†	**0.3064**†	**0.4315**†	**0.2435**†	**0.2391**†	**0.2102**†

used to expand the query. The number of feedback terms for our proposed *ProposedQE* approach, top-$\mathcal{N}$ was set to 20. The sensitivity of this choice depicted in Figure 5.5. In addition, the sensitivity of our *ProposedQE* method to the anchoring parameter α depicted in Figure 5.6. According to this figure, the *ProposedQE* method obtained the best performance when α was set to 0.8 for both the TMB2011 and TMB2012 test set. Here, we reported the results based on these settings.

Results showed that both the *TMQE* and *ProposedQE* methods significantly ($p < 0.05$) outperform the baseline for the *allrel* relevant criteria in terms of all evaluation measures on both TMB2011 and TMB2012 test set. Similarly, for the *highrel* criteria, both methods significantly outperform the baseline in terms of P@30 for the TMB2011 test set and in terms of all evaluation measures for

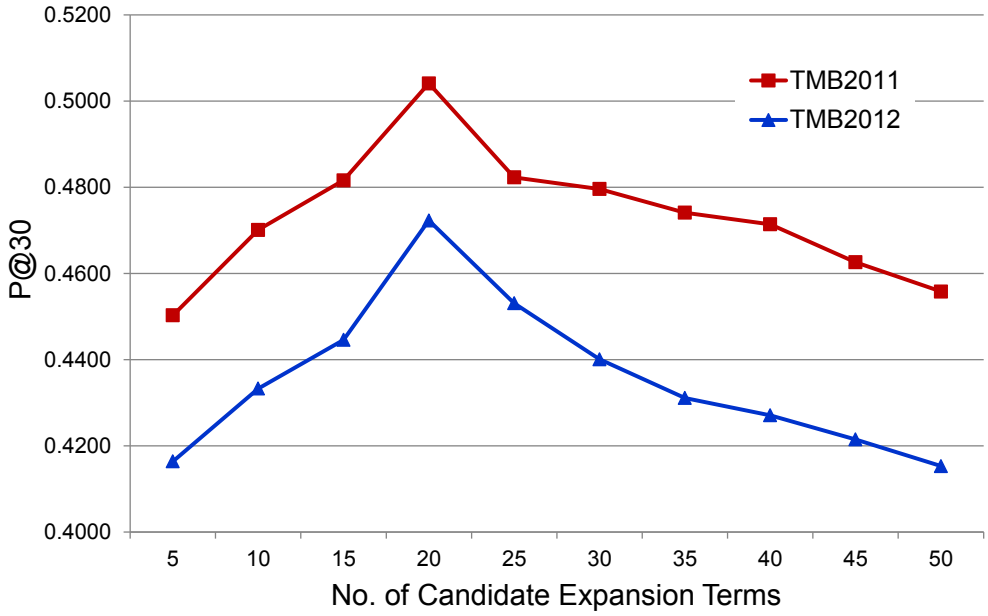

Figure 5.5: Effect of the increasing number of candidate expansion terms, $\mathcal{N}$ on TMB2011 and TMB2012 test set.

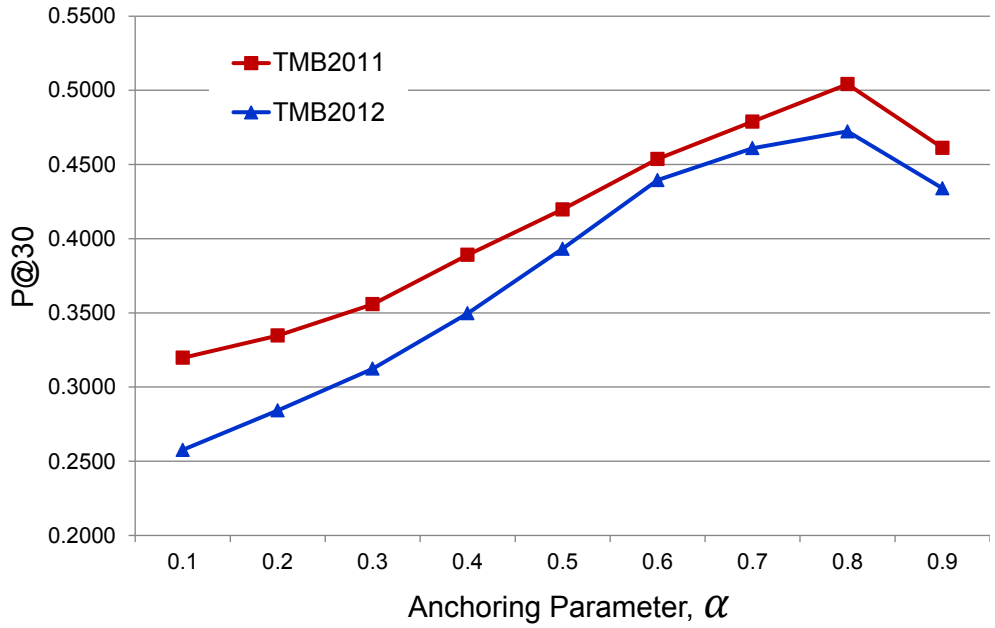

Figure 5.6: Sensitivity of the ProposedQE method to the anchoring parameter, α.

the TMB2012 test set. Though for the *highrel* criteria, the *ProposedQE* method obtained significantly indistinguishable performance for the TMB2011 test set in terms of R-Prec and MAP measures, it outperforms the baseline by a small margin. This observation validates the effectiveness of our *ProposedQE* method for selecting good expansion terms to reformulate the query which in turn enhance the performance of microblog retrieval.

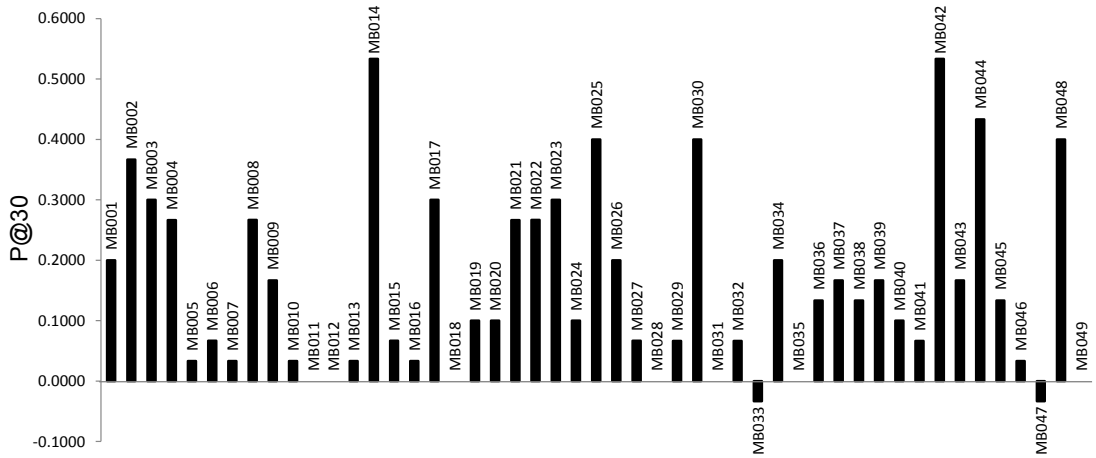

Figure 5.7: Query-wise performance analysis (TMB2011 query set). The increase(+) / decrease(-) of the P@30 of ProposedQE method compared to the baseline.

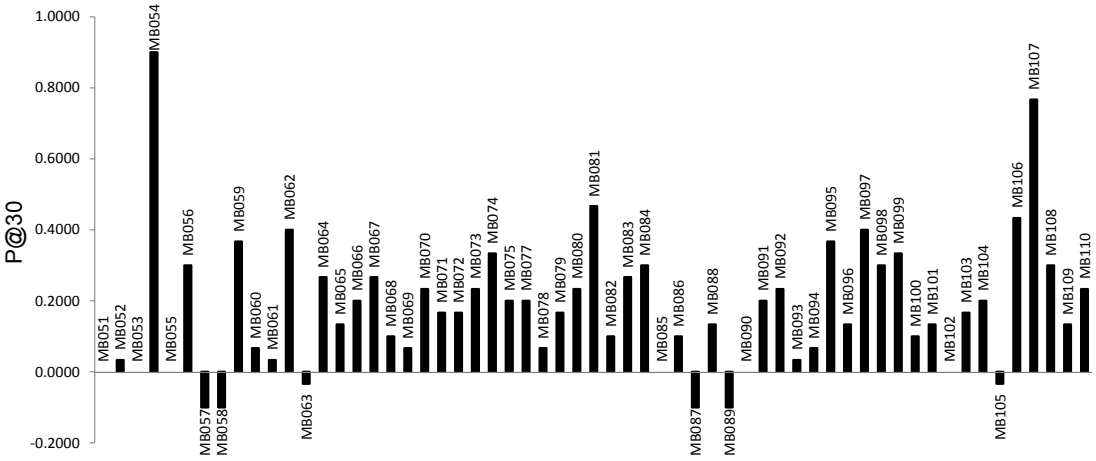

Figure 5.8: Query-wise performance analysis (TMB2012 query set). The increase(+) / decrease(-) of the P@30 of ProposedQE method compared to the baseline.

Though our query expansion methodology leads to a better result, we performed the query-wise analysis to understand how many queries are benefited by our system and how many are hurt. Figure 5.7 and Figure 5.8 illustrated the query-wise improvement of our *ProposedQE* method over the baseline for *allrel* relevant criteria based on individual test queries of TMB2011 and TMB2012 test set, respectively. According to these figures, it is observed that our system obtained significant improvement by a large margin for some of the queries and obtained the moderate improvement for most of the queries. Whereas, for some other queries, the improvement is relatively small. However, our system lags in 8 queries, where 2 queries are from TMB2011 and 6 queries are from TMB2012 test set.

5.3.3 Feature Analysis

To understand the effectiveness of our several feature categories used for ranking candidate terms, including lexical and term distribution based features, twitter specific features, temporal features, sentiment aware features, and embedding based features, we evaluated the performance of each group with a feature ablation study by utilizing the TMB2011 test collection. In this regard, we removed one feature group each time and repeated the experiment. Results of these experiments were illustrated in Figure 5.9.

In Figure 5.9, it can be observed that precision at top G tweets (P@G) drops substantially for several feature categories. For the simplicity of discussion, we considered G=30 (i.e. P@30) and compared the difference in results while removing each feature group. We have seen that when removing temporal features the graph point drops to a large extent and the difference in results with the all features group is statistically significant at ($p < 0.05$). This deduced the importance of our proposed temporal features for selecting temporally relevant effective expansion terms. We also observed the significant ($p < 0.05$) decrease in the result while removing our proposed embedding based features, which revealed the importance of considering semantic relatedness between the query and candidate terms during the expansion terms selection. Along with this direction, removing lexical and term distribution based features also lead to a significant ($p < 0.05$)

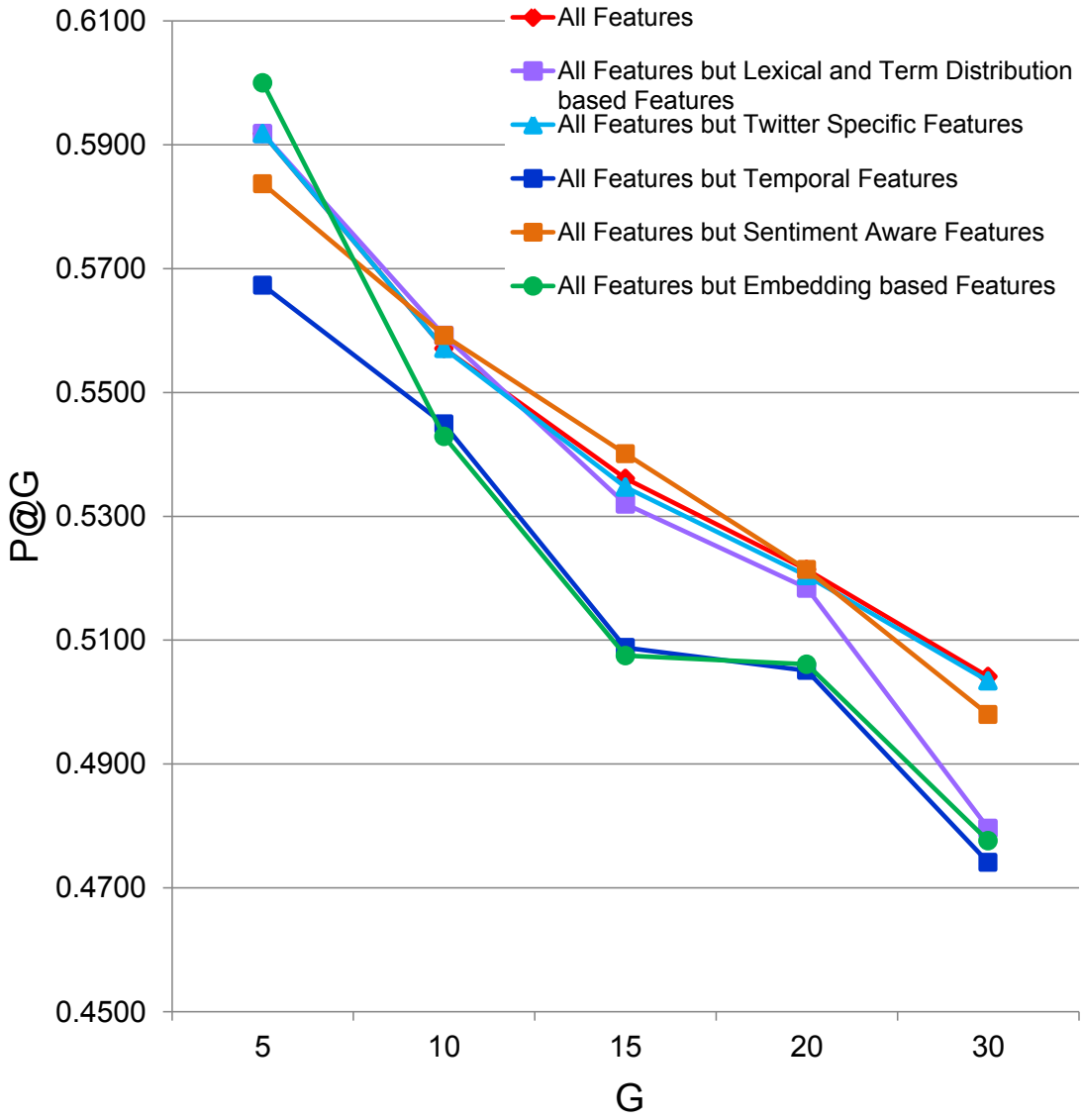

Figure 5.9: P@G performance with different feature categories.

decrease in precision, which deduced the importance of these traditional features. Similarly, the decrease in performance is significant at ($p < 0.05$) when removing the sentiment aware features, thus deduced the importance of considering sentiment aspect for good candidate terms selection. However, while removing the twitter specific feature, the performance decreases slightly though the difference is not significant. This is because our twitter specific feature cannot discriminate the good and bad expansion terms effectively.

5.3.4 Comparison with Related Work

We compared the performance of our proposed query expansion (ProposedQE) method with some of the competitive related methods including Chy et al. [47], Miyanishi et al. [19], Kuzi et al. [40], Rao and Lin [18], Albishre et al. [15], Lavrenko and Croft [187], and Rocchio et al. [188]. We implemented all of these query expansion methods in our retrieval framework but Rao and Lin [18] and Miyanishi et al. [19]. This is because similar to our method, these two methods used the language model with Dirichlet smoothing as the retrieval model. The results based on the TREC Microblog 2011 and 2012 test collections were presented in Table 5.5 and Table 5.6, respectively.

Table 5.5: Comparative performance (P@30, MAP, and NDCG@30; higher is better) with other methods on TMB2011 test set. The best results are highlighted in boldface. † indicates the statistically significant difference between our proposed method (ProposedQE) and the other methods at ($p < 0.05$).

Method	Allrel			Highrel	
	P@30	MAP	NDCG@30	P@30	MAP
Baseline	0.3483†	0.3050†	0.4374†	0.1253†	0.2378
Albishre et al. [15] (ACSW,17)	0.3605†	0.3069†	0.5444	0.0782†	0.1283†
Rocchio et al. [188]	0.3891†	0.3703†	0.4693†	0.1131†	0.2178
Kuzi et al. [40] (CIKM,16)	0.4122†	0.3770†	0.4975†	0.1313†	0.2521
Chy et al. [47] (IEICE TOIS,17)	0.4143†	0.3750†	0.4819†	0.1455†	0.2251
LSIQE [137]	0.4204†	0.3599†	0.4747†	0.1404†	0.2220
Lavrenko and Croft [187] (SIGIR,01)	0.4286†	0.3812†	0.4896†	0.1404†	0.2534
Rao and Lin [18] (ICTIR,16)	0.4388†	0.4024	-	-	-
Miyanishi et al. [19] (ECIR,13)	0.4830	0.2741†	-	-	-
Our Proposed Method ((ProposedQE))	**0.5041**	**0.4222**	**0.5518**	**0.1798**	**0.2554**

From Table 5.5, the results showed the significant improvements over all the query expansion methods ([15, 18, 47, 40, 187, 188], and baseline) but [19], in terms of primary evaluation measure P@30 for both the *allrel* and *highrel* criteria. Similarly, from Table 5.6, the results showed the significant improvements over all the methods ([15, 18, 47, 40, 187, 188], and baseline). We also reported the results in terms of other evaluation measures.

Miyanishi et al. [19] proposed a time-based query expansion (QE) method

Table 5.6: Comparative performance (P@30, MAP, and NDCG@30; higher is better) with other methods on TMB2012 test set. Legend settings are identical to Table 5.5.

Method	Allrel			Highrel	
	P@30	MAP	NDCG@30	P@30	MAP
Baseline	0.2932†	0.1815†	0.2862†	0.1625†	0.1388†
Albishre et al. [15] (ACSW,17)	0.3232†	0.2002†	**0.4623**	0.1740†	0.1544†
Rao and Lin [18] (ICTIR,16)	0.3514†	0.2325†	-	-	-
LSIQE [137]	0.3638†	0.2421†	0.3414†	0.2054†	0.1687†
Rocchio et al. [188]	0.3667†	0.2265†	0.3591†	0.2113	0.1828†
Chy et al. [47] (IEICE TOIS,17)	0.3751†	0.2718†	0.3630†	0.1923†	0.1860†
Kuzi et al. [40] (CIKM,16)	0.3797†	0.2567†	0.3558†	0.1970†	0.1779†
Lavrenko and Croft [187] (SIGIR,01)	0.4068†	0.2658†	0.3709†	0.2101†	0.1845†
Our Proposed Method (ProposedQE)	**0.4723**	**0.3064**	0.4315	**0.2435**	**0.2102**

that can handle the recency and temporal variation according to the topic's temporal variation. Whereas Rao and Lin [18] utilized the continuous hidden Markov model (cHMM) to identify the bursty temporal clusters where tweets in the bursty states were selected for query expansion. For scoring terms, they only considered the query likelihood scores. But the above two methods only consider the temporal aspect to select the candidate terms although some queries are temporally insensitive [47]. However, our proposed temporal relatedness approach for candidate tweets selection, recency, and time series based temporal features effectively addressed the temporality of the query-term pair. In addition, we have devised a rich set of lexical, embedding, and sentiment aware features to estimate the term relatedness. Therefore, our method effective for both the temporal and non-temporal queries. Albishre et al. [15] combined the lexical and latent Dirichlet allocation (LDA) based topical evidence from pseudo-relevance feedback (PRF) into their discriminative expansion approach to meet the user interests. However, LDA might not work well to uncover the hidden topics from the noisy short texts like tweets [138]. In contrast, we have utilized the biterm topic model (BTM) [138] in our proposed framework, which is effective for modeling topics in tweets. Chy et al. [47] proposed a three-stage query expansion technique. They utilized the pseudo-relevant tweets at the first stage, made use of Web search

results at the second stage, and extracted hashtags relevant to the query at the third stage. For weighting terms, they used the IDF-score of each term. However, only the IDF-score based term scoring method might induce irrelevant rare terms from the noisy tweet contexts. In contrast, we have utilized several term weighting schemes under the supervised manner to quantify the term relatedness without utilizing any external resources such as Web search results. Therefore, our method effectively eliminates the irrelevant terms during the candidate terms selection and obtained better result compared to [47]. Kuzi et al. [40] leveraged the centroid-based representation of the embedding vectors of the query terms to select the semantically related candidate terms for query expansion. However, our embedding based features effectively estimate the semantic relatedness between the candidate terms and query.

Along with this direction, we employed a query expansion strategy for comparison where latent semantic indexing (LSI) [137, 197] was used to select the candidate terms. We denoted the experimental setting as LSIQE. The LSIQE procedure was based on two steps: (i) dimensionality reduction of term-by-tweet matrix generated from top retrieved PRF tweets by singular value decomposition (SVD) because similar terms tend to be closer in lower dimensional space and (ii) estimate the cosine similarity between the query vector and term-vector in the space of terms. Based on the similarity score, candidate terms were selected to expand the query.

Moreover, we compared the performance of our method against the state-of-the-art PRF based query expansion models RM3 (Lavrenko and Croft [187]) and Rocchio (Rocchio et al. [188]). The basic idea of RM3 is to estimate the relevance feedback using relevance models such as query likelihood. Previous studies (e.g. [187, 198]) already demonstrated the robustness of RM3 model against a number of state-of-the-art query expansion methods. On the other hand, the Rocchio [188] model incorporates the pseudo-relevant information into the vector space model (VSM), where unique terms of the pseudo-relevant tweets set are ranked in a descending order of their TF-IDF weights. In summary, the results in Table 5.5 and Table 5.6 demonstrated the superiority of our method for selecting effective expansion terms for query expansion in microblog retrieval.

5.3.5 Discussion

To demonstrate the effectiveness of our proposed temporal relatedness (TR) approach for candidate tweet selection, we compared the performance of our *ProposedQE* method with and without using this approach. The results were presented in Table 5.7 and Table 5.8 for the TMB2011 and TMB2012 test set, respectively. It showed that excluding TR approach, the performance decrease significantly ($p < 0.05$) in terms of P@30, MAP, and NDCG@30 evaluation measures. Therefore, we can deduce that our TR approach effectively selects the pseudo-relevant tweets that boost the method with relevant expansion terms. However, we also reported the results while excluding the PRF approach.

Table 5.7: Performance comparison of our method with/without each candidate tweet selection approach (PRF and temporal relatedness (TR)) on TMB2011 test set. The best results are highlighted in boldface. † indicates statistically significant difference at ($p < 0.05$) between ProposedQE and other methods.

Method	Allrel			
	P@30	R-Prec	MAP	NDCG@30
ProposedQE	**0.5041**	**0.4301**	**0.4222**	**0.5518**
Without PRF	0.4578†	0.4105	0.4060	0.5296†
Without TR	0.4707†	0.4156	0.4117†	0.5265†

Table 5.8: Performance comparison of our method with/without each candidate tweet selection approach (PRF and temporal relatedness (TR)) on TMB2012 test set. Legend settings are identical to Table 5.7.

Method	Allrel			
	P@30	R-Prec	MAP	NDCG@30
ProposedQE	**0.4723**	**0.3615**	**0.3064**	**0.4315**
Without PRF	0.4192†	0.3483	0.2886†	0.3898†
Without TR	0.4147†	0.3431†	0.2860†	0.3841†

Table 5.9: Examples of top-10 expansion terms extracted by our ProposedQE method and three other competitive query expansion methods. Boldfaced terms are relevant to the respective query.

Original Query	Methods	Extracted Expansion Terms
2022 FIFA soccer	ProposedQE	**qatar cup world winter bl-atter sepp president stadiums** tune changes
	Lavrenko & Croft [187]	south dice **winter** 09 2010 **summer** se stvy #futbol **world**
	Rocchio et al. [188]	11 **cup blatter** playing **wo-rld qatar sepp** stvy **win-ter** plans
	Chy et al. [47]	11 play **qatar world cup** governing move **host** mar dec
The Daily	ProposedQE	**newspaper ipad corp lau-nch** finally **rupert** media **murdoch launches future**
	Lavrenko & Croft [187]	**newspaper** installed 01 02 sh-ow 2011 privacy azeroth #apple world
	Rocchio et al. [188]	chest out free baked planet allbritishsupermancasting i-ntroduzione versicherungen delphiusa andrewcsfan
	Chy et al. [47]	**ipad** 2011 **newspaper news** amp coffee latest world reporting charleston university
Egyptian evacuation	ProposedQE	**americans turkey embas-sy travel military citi-zens states united cairo voluntary**
	Lavrenko & Croft [187]	**ohio updates foreign katrina leak gas fire state detroit plan**
	Rocchio et al. [188]	centre cyclone **cairo** tcyasi **flight gas** cairns **leak yasi begins**
	Chy et al. [47]	**gas leak egypt** feb **citizens** libya **evacuate** showing **gaza border**
Hugo Chavez	ProposedQE	**president venezuelan enemy #golf golf venezuelas** efe **threatened** seize **bank**
	Lavrenko & Croft [187]	#freeve role office seek **ye-ars venezuela** seguindo los se **bank**
	Rocchio et al. [188]	**venezuelan president ven-ezuelas marks** seek term third six dictator 12
	Chy et al. [47]	boss office dictator egypt author crisis new **president venezuelan venezuela**

For qualitative analysis, we have enlisted four example queries along with the top-10 candidate expansion terms extracted by our *ProposedQE* method and three related methods including RM3 [187], Rocchio [188], and the model proposed by Chy et al. [47] in Table 5.9. Boldfaced terms are relevant to the respective query. The relevancy was estimated based on our term labeling strategies described in Section 5.2.6. The terms were ordered by their rank score. As a

specific example, if we took the query "2022 FIFA soccer," we see that eight out of ten extracted terms by our *ProposedQE* method are relevant. On December 2010, Qatar won the right to host the World cup, 2022. Due to the hot weather of Qatar, later on January 25 2011, FIFA president Sepp Blatter said that "2022 World Cup could be held at the end of the year" that means during the winter. After that, lots of people posted their opinions on twitter about this issue. Therefore, we can deduce that extracted expansion terms by our *ProposedQE* method are relevant to the query and can alleviate the vocabulary mismatch problem effectively compared to other methods. This observation validates the effectiveness of our proposed method.

Besides the qualitative analysis of the expansion terms, we took two example queries including "The Daily" and "organic farming requirements" and compared their retrieval effectiveness towards the baseline rank of the retrieved tweets. The corresponding results were presented in Table 5.10 and Table 5.11, respectively. From Table 5.10, we see that all the top 10 retrieved tweets are relevant to the query "The Daily" and the baseline rank of these tweets are very far from their current rank. This observation validates that our proposed query expansion method effectively retrieves the relevant tweets in compared to the baseline. In addition, the quality of these tweets also demonstrates the usefulness of our query expansion method. However, from Table 5.11, we see that our expanded query only retrieve one relevant tweet for the query "organic farming requirements". Here, we see that our expanded query retrieved some tweets (e.g. rank# 3, 4, 5, 8, 10) that seem to be relevant to the query. But the assessor didn't annotate them relevant. Other methods also didn't achieve satisfactory results for such worst queries.

Table 5.10: Successful example of tweet retrieval using the expanded query.

Tweet ID	Current Rank	Retrieved Tweets for the query "The Daily"	Baseline Rank
$32 \cdots 12$ ▶	1	VentureBeat: News Corp has spent $30M on The Daily iPad newspaper: Rupert Murdoch is finally launching The Daily... URL	78
$32 \cdots 48$ ▶	2	News Corp Has Spent $30M on The Daily iPad Newspaper - Rupert Murdoch is finally launching The Daily today, but ther... URL	106
$33 \cdots 76$ ▶	3	Rupert Murdoch's much anticipated iPad newspaper, The Daily, officially launched yesterday at a daily rate of .14/day	131
$32 \cdots 04$ ▶	4	Murdoch's iPad newspaper to be launched (AFP): AFP - News Corp.'s Rupert Murdoch is to unveil "The Daily" o... URL	>1000
$32 \cdots 88$ ▶	5	The Daily iPad 'newspaper' launches, $.99 weekly or $39.99 per year: Rupert Murdoch's iPad-... URL #ipad #iphone #apple	>1000
$32 \cdots 56$ ▶	6	The Daily: a review of Murdoch's iPad newspaper: Rupert Murdoch's iPad launch the Daily offers glitzy graphics, ... URL	107
$32 \cdots 33$ ▶	7	IPad newspaper The Daily launches its first edition: News Corp. CEO Rupert Murdoch and a gaggle of tech and medi... URL	>1000
$30 \cdots 37$ ▶	8	#money Murdoch's iPad newspaper The Daily to launch Feb. 2 - News Corp.'s iPad-exclusive newspaper, The Daily, will ... URL	96
$32 \cdots 61$ ▶	9	The Daily to launch later today on iPad - Rupert Murdoch is set to launch The Daily, a virtual paper exclusive to th... URL	76
$32 \cdots 96$ ▶	10	The Daily iPad 'newspaper' launches: Rupert Murdoch's iPad-only magazine _The Daily_, once described as "The New... URL	46

Table 5.11: Unsuccessful example of tweet retrieval using the expanded query.

Tweet ID	Current Rank	Retrieved Tweets for the query "organic farming requirements"	Baseline Rank
$32 \cdots 20$ ▶	1	"Organic eggs set for testing times" - Farming UK: URL - we must try and help organic producers wherever possible	14
$33 \cdots 48$	2	USDA/Vilsack decision on GM alfalfa WILL HURT ORGANIC FARMING in this country!	22
$33 \cdots 96$	3	Ramsey Bros have a wide range of industry leading equipment to suit your farming requirements with branch... URL	16
$32 \cdots 20$	4	OSU Well Represented at Ohio Organic Farming Conference: "Nematodes as Monitoring Tools for Soil Foodweb Health ... URL	32
$33 \cdots 10$	5	Organic Farming Changes Everything for a Community in India — Gaiam Life URL	28
$30 \cdots 20$	6	@kellyjanice Organic farms cannot produce enough food to feed the country. Monsanto crops are vital to the (real) farming that feeds the US.	9
$32 \cdots 68$	7	Farming : New Congress shows hostility to organic farming — Rodale Institute URL	4
$29 \cdots 12$	8	Vt. organic farming group prepares for conference URL	19
$32 \cdots 82$	9	Is organic farming policy-driven or consumer-led? URL	20
$29 \cdots 68$	10	Interested in Organic Farming? Then do one of these cou... - URL	24

5.4 Summary

In this chapter, we presented a query expansion framework focusing on an ensemble of features to mitigate the vocabulary mismatch problem in microblog retrieval. Upon improving the performance of the baseline retrieval with our proposed topic modeling based query expansion, we introduced a temporal relatedness approach based on C-LSTM, which effectively selects the candidate tweets from the temporal area where a query topic is actively mentioned. We introduced several temporal features to estimate the temporal association between a candidate expansion term and query terms. We also proposed several embedding based features to select the semantically related candidate term with respect to the query terms. In addition, we also proposed and utilized some lexical, twitter specific, and sentiment aware features to quantify the relatedness of a candidate term. To select the best set of features, we employed the elastic-net regularization as a supervised feature selection method. Based on the selected features, a linear learning to rank (L2R) model was used to estimate the relevance score of candidate terms. Experimental results on the TREC microblog dataset demonstrated that our proposed method outperformed some competitive query expansion methods.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we have focused on the research problems related to microblog retrieval. In this regard, we have proposed a reranking based approach and a query expansion based approach, where several novel techniques are introduced to improve the retrieval effectiveness. We summarize the key contributions of our thesis in the following:

- **Temporal Features:** To tackle the challenges of real-time nature of the twitter, we have proposed several temporal features for both the reranking and query-expansion based approaches. By utilizing the query time and tweet time, we have proposed recency score and burst-aware score feature to estimate the temporal relevance in our reranking approach. Whereas, to select the temporally relevant expansion terms in our query expansion approach, we have proposed the time series similarity features and temporal distance feature by leveraging the temporal correlation of candidate term and query terms.
- **Contextual Features:** To estimate the semantic relevance between the query-tweet pair in the reranking approach and query-term pair in the query expansion approach, we have introduced several contextual features by leveraging the word embedding, kernel density estimation, and sentiment correlation of the query-tweet and query-term pair. Experimental results demonstrated the effectiveness of our proposed features.

- **Twitter Specific Features:** Besides the temporal and contextual features, we have introduced several features in our proposed reranking and query expansion approaches by exploiting the several twitter characteristics. Our proposed twitter specific features include the hashtag feature, hashtag importance features, tweet popularity feature, query terms in URL feature, and URL popularity feature.
- **Query Type Classification:** To determine the queries temporal and sentiment sensitivity in our reranking based approach, we introduce a query type determination technique by leveraging the temporal and sentiment distribution of the top retrieved tweets.
- **Temporal Relatedness (TR) Approach for Candidate Tweet Selection:** To generate the pool of effective candidate terms, we have introduced a temporal relatedness (TR) approach based on C-LSTM for candidate tweet selection. Experimental results demonstrated that our proposed TR approach effectively selects the temporally-relevant tweets that boosts the query expansion method with relevant expansion terms.
- **Topic Modeling based Query Expansion:** To improve the performance of the baseline retrieval model in our query expansion framework, we introduce an effective topic modeling based query expansion (TMQE) technique, where candidate terms are ranked based on their topic coverage i.e. terms that represent the maximum number of topics get the highest rank.
- **Summary of Experiments and Results:** To evaluate the performance of our proposed reranking and query expansion approaches, we have conducted experiments on TREC Microblog 2011 and 2012 test collections over the TREC Tweets2011 corpus. Experimental results demonstrated the effectiveness of our method over the baseline and known related works in terms of several evaluation measures and relevance criteria.

In the remainder of this chapter, we will discuss some issues related to our approaches and some of the possible future research directions.

6.2 Future Directions

In the future, we have a plan to incorporate the microblog specific quality indicators for estimating the quality of the tweet and candidate terms. In our query expansion framework, we have limited our investigation to a linear learning to rank (L2R) model to rank the candidate terms. Future studies may investigate more efficient and effective L2R algorithms for candidate term ranking. Moreover, we have a plan to introduce some graph-based methods by leveraging the user-user, user-tweet, user-location, and tweet-location correlation to estimate the relevance of query-tweet and query-term pair. An overview of our intended future works are presented below:

Exploiting Social Graph for Microblog Retrieval

In microblog, a user follows the other user if they belong to the same interest domain. For example, a researcher follows the other researchers in the same field. Therefore, when a tweet is posted by a researcher it will be usually discussed within the community of this field. We hypothesize that it might be an important clue who like and retweet a posted tweet as well as who are the follower of this tweeter (i.e. tweet author). If we exploit this characteristic of twitter to generate the user-tweet and user-user social graph and applying the state-of-the-art deep learning technologies to extract some features that might be an important clue to estimate the relevancy of query-tweet pair as depicted in Figure 6.1.

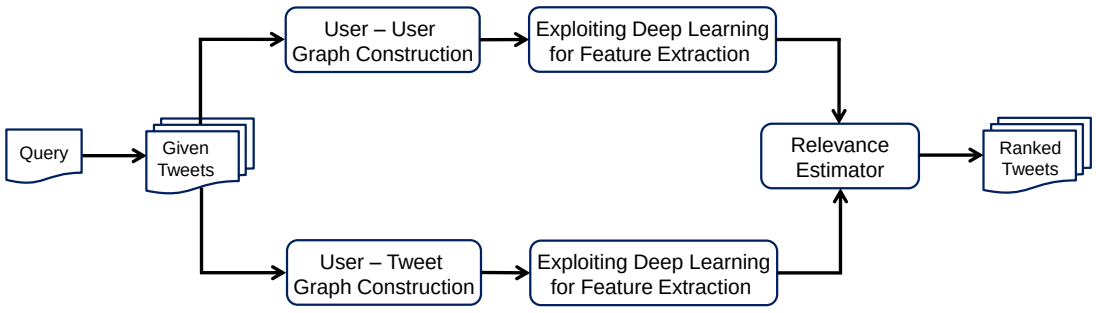

Figure 6.1: Exploiting social graph for relevance estimation.

Exploiting Location Graph for Microblog Retrieval

People usually share the location a lot in their posted tweets when a notable event occurs in a specific place and during the disaster period. Since location information helps us to associate the tweet with the actual physical location, it might be an important clue to estimate the relevancy. For example, when the tsunami occurred in Japan in 2011, tweets that are posted near the affected area might be more important to serve the situational information need than the huge amount of condolence tweet generated from the outside world. Therefore, exploiting user-location and tweet-location graph to extract the relevance signal (as depicted in Figure 6.2) might be useful for microblog retrieval. But in developing countries, geo-tagged tweets are very sparse, therefore extracting location information from tweet text would be another important research direction.

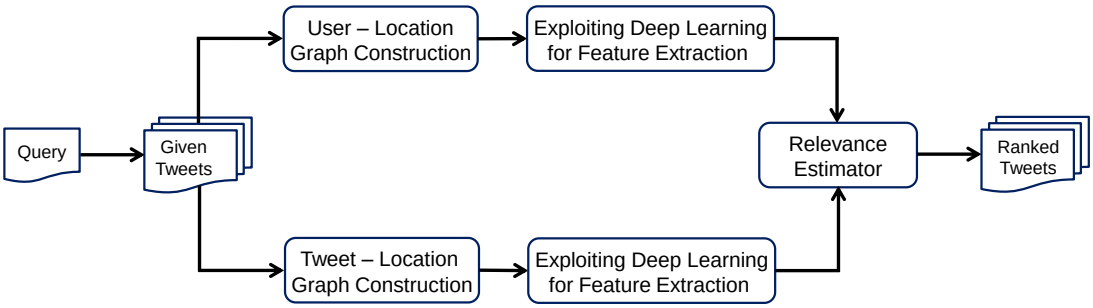

Figure 6.2: Exploiting location graph for relevance estimation.

Categorization of Queries for Microblog Retrieval

In our tweet reranker framework, we categorize the queries based on temporal and sentiment sensitivity. Our experimental results demonstrated that categorization of queries significantly improve the performance of microblog retrieval, which in turn established its significance. Therefore, exploiting the different types of query categories (e.g. cyclic queries, trending queries, contextual queries, etc.) for microblog retrieval would be another future direction.

Related Publications

Articles in International Journals:

- **Chy, A. N.**, Ullah, M. Z., Aono, M. : Microblog Retrieval Using Ensemble of Feature Sets through Supervised Feature Selection, IEICE Transactions on Information and Systems (IEICE TOIS), Vol. 100, No. 4, pp. 793806, 2017, DOI: 10.1587/transinf.2016DAP0032

- **Chy, A. N.**, Ullah, M. Z., Aono, M. : Query Expansion for Microblog Retrieval Focusing on an Ensemble of Features, Journal of Information Processing (JIP), Vol. 27, pp. 61-76, 2019, DOI: 10.2197/ipsjjip.27.61

Articles in International Conferences:

- **Chy, A. N.**, Ullah, M. Z., Aono, M. : Combining Temporal and Content Aware Features for Microblog Retrieval, IEEE International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA 2015), pp. 1-6, August 19-22, Chonburi, Thailand, 2015,
  DOI: 10.1109/ICAICTA.2015.7335353 (**Best Student Paper Award**)

- Ullah, M. Z., Shajalal, M., **Chy, A. N.**, Aono, M. : Query Subtopic Mining Exploiting Word Embedding for Search Result Diversification, Twelfth Asia Information Retrieval Societies Conference (AIRS 2016), pp. 308-314, November 30-December 2, Beijing, China, 2016, DOI: 10.1007/978-3-319-48051-0_24 (**Best Presentation Award**)

- Shajalal, M., Ullah, M. Z., **Chy, A. N.**, Aono, M. : Query Subtopic Diversification based on Cluster Ranking and Semantic Features, IEEE In-

ternational Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA 2016), 6 pages, August 13-16, Penang, Malaysia, 2016, DOI: 10.1109/ICAICTA.2016.7803099

- Siddiqua, U. A., **Chy, A. N.**, and Aono, M. : Stance Detection on Microblog Focusing on Syntactic Tree Representation, 3rd International Conference on Data Mining and Big Data (DMBD 2018), pp. 478-490, Shanghai, China, 2018, DOI: 10.1007/978-3-319-93803-5_45

Workshop/Technical Papers:

- **Chy, A. N.**, Ullah, M. Z., Aono, M. : A Time and Context Aware Reranker for Microblog Retrieval, The 29th Annual Conference of the Japanese Society for Artificial Intelligence, May 30 - June 2, Hokkaido, Japan, 2015.

- **Chy, A. N.**, Ullah, M. Z., Shajalal, M., Aono, M. : KDETM at NTCIR-12 Temporalia Task: Combining a Rule-based Classifier with Weakly Supervised Learning for Temporal Intent Disambiguation, Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR, pp. 281-284, June 7-10, Tokyo, Japan, 2016.

- **Chy, A. N.**, Siddiqua, U.A., Aono, M. : Neural Networks and Support Vector Machine based Approach for Classifying Tweets by Information Types at TREC-2018 Incident Streams Task, Proceedings of the 27th Text REtrieval Conference (TREC), pp. –, November 14-16, NIST, USA, 2018.

Appendix A

Experiments with Microblog Retrieval during Disasters

A.1 Introduction

Microblog platforms such as twitter, tumblr, sina weibo, etc. are rapidly moving towards a platform for mass collaboration in user-generated information production. Twitter has become the most popular among the microblog services. Everyday lots of people turning to this online platform to share their views, opinions, breaking news as well as fulfill their diverse information needs. The real-time nature of the twitter plays an important role during a disaster period, such as earthquake, floods, wildfires, and typhoons. Because the user-generated twitter posts during such events might be useful to serve the situational information needs [4]. However, due to the brevity of the tweets and noisy tweet contents, information retrieval in twitter is regarded as a challenging IR problem. To address the general real-time information seeking behaviors, TREC was introduced the microblog ad-hoc search task in 2011 [6]. In contrast, this year TREC-2018 introduces an incident streams (TREC-IS) task designed specifically to tackle the microblog retrieval during a disaster period. The main task for the 2018 TREC-IS track was to categorize the tweets in each event/incident's stream into different high-level information types defined in the TREC-IS incident ontology.

In this chapter, we proposed our approaches to address the challenges of 2018 TREC-IS task. We combine different types of classifiers in our proposed approaches. We define a set of rules for the rule-based classifier based on the lan-

guage of tweets, exploiting indicator terms available in the training corpus, and WH-orientation of tweets. We consider lexical and content relevance features, incident and event related features, twitter specific features to train our multi-class SVM classifier, whereas a pre-trained *word2vec* model is used for the deep neural network (DNN) based classifiers.

The rest of the contents are structured as follows: We will introduce our proposed framework in **Section A.2**. **Section A.3** includes experiments and evaluation to show the effectiveness of our proposed methods. Some concluded remarks and future directions of our work described in **Section A.4**.

A.2 Proposed Approach

Now, we describe the details of our proposed framework. Given a query related to an event/incident and a set of tweets, the goal of our proposed system is to categorize the tweet into the high-level information types. The overview of our proposed framework depicted in Figure A.1.

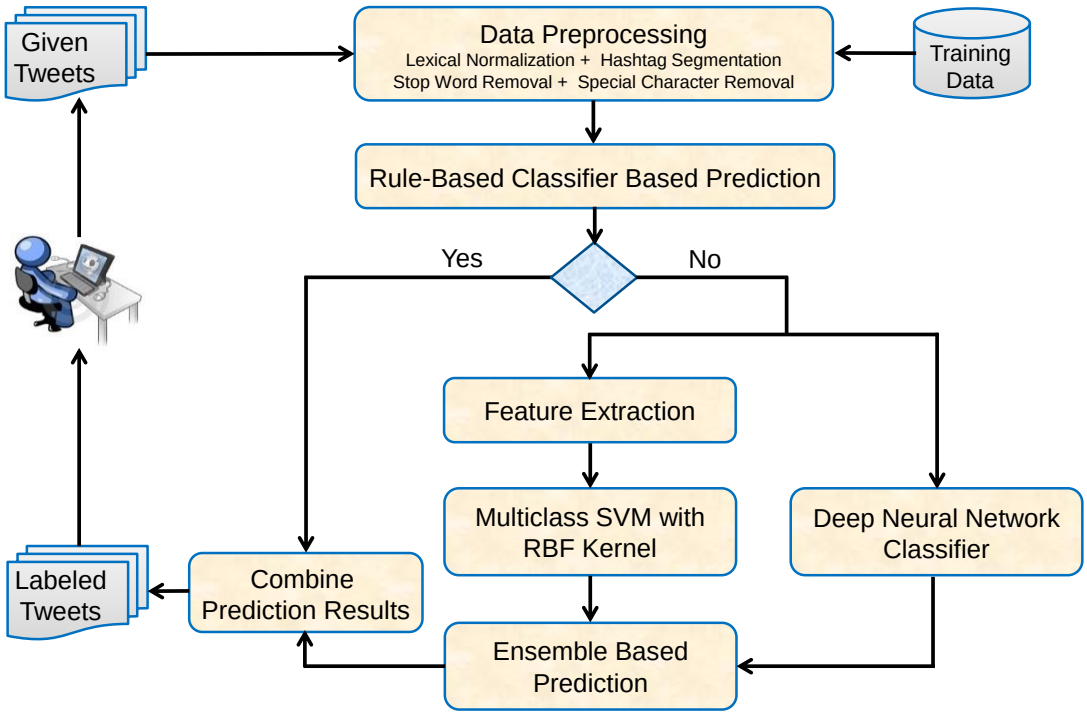

Figure A.1: Proposed TREC incident streams (TREC-IS) system.

At first, our system fetches a query and the corresponding tweet set as a single batch and indexed them for further processing. In the data preprocessing stage, we perform the tokenization, lexical normalization to the tokenized words, stop-word removal, special character removal, and hashtag segmentation. Next, our proposed rule-based classifier is applied to classify the tweets into the corresponding high-level information types. For the tweets that are not classified by the rule-based classifier, we consider the combined weighted prediction score from multi-class support vector machine (SVM) classifier and several deep neural network (DNN) classifiers. SVM classifier is trained with our extracted features. We extract several effective features broadly grouped into four different categories, including lexical and content relevance features, incident and event related features, twitter specific features, and sentiment aware features. For extracting sentiment aware features, we construct strong sentiment lexicons by combining several publicly available sentiment lexicons. To scale the feature value, we make use of the *Min-Max* normalization technique. For DNN based classifiers, a pre-trained *word2vec* model is applied. Tweets are labeled to the information type that gets the highest prediction score. Results of both the rule-based classifier and the ensemble of classifiers are then combined and the set of labeled tweets return to the user.

A.2.1 Dataset Preprocessing

Data preprocessing stage is initiated with tokenization. As tweets are informal user generated contents, people use lots of non-english characters and symbols in tweets. Since meaningful English words do not contain these characters, we remove these characters from tweets. Moreover, the short length constraint of the tweet makes characters expensive. To overcome this constraint, people are utilizing twitter specific syntaxes such as #hashtag to express their thoughts concisely. For #hashtag removal, we segment each #hashtag by using a hashtag segmentation technique similar to Siddiqua et al. [199] and replaced the hashtag with the segmented words.

Moreover, tweets often contain non-standard word forms and domain-specific entities. For example, people usually use “earthquakeeeee” instead of “earth-

quake," "addiquate" instead of "adequate," "appt" instead of "appointment," etc. We utilized two lexical normalization dictionaries collected from [167] and [168] to normalize such non-standard words into their canonical forms. In incident streams task, stopwords play a negative role because they do not carry any incident-oriented information and may actually damage the performance of the classifiers. For stopword removal, we applied the Indri's standard stoplist[1].

A.2.2 Rule-based Classifier

In rule-based classifiers, we usually construct a set of rules that determine a certain combination of patterns, which are most likely to be related to the different classes or information types. Each rule consists of an antecedent part and a consequent part. The antecedent part corresponds to the word patterns and the consequent part corresponds to a class label. We can define a rule as follows:

$$R_j : \text{if } x_1 \text{ is } A_{j1} \text{ and } ..... \ x_n \text{ is } A_{jn}$$
$$\text{then } Class = C_j, \quad j = 1, ..., N$$

where R_j is a rule label, j is a rule index, A_{j1} is an antecedent set, C_j is a consequent class, and N is the total number of rules.

Our unsupervised rule-based classifier casts the TREC incident streams task as a multi-class classification problem and labeled each tweet to the corresponding information types assigned by the rules. To achieve this, we define a set of rules based on the tweets language, indicator terms within tweets, and WH-orientation of the tweet. Descriptions of each set of rules are presented next.

A.2.2.1 Language Related Rule

Even though twitter is a multilingual microblog service, we only consider English tweets as relevant in this research. Therefore, we define a rule based on the language of a tweet that is if the language of a tweet is not English, we classify the tweet as *Irrelevant* information type. To identify the non-English tweets from the given tweet set, a language detection library [200] was applied in our system.

[1]https://www.lemurproject.org/stopwords/stoplist.dft

A.2.2.2 Indicator Terms based Rule

A tweet may contain some highly influential indicator terms related to a high-level information type which may useful to categorize the tweet into the corresponding information type. We exploit the indicator terms given in the training data to prepare two curated indicator terms lexicon. One for the *MultimediaShare* category and one for the *Donations* category. If a tweet contains words from these lexicons, it is classified to the corresponding information type. The priority of the information type determined by the number of lexicon words that the tweet contains.

A.2.2.3 WH-Orientation based Rule

Since people usually use WH sentence to know more about the incident, we use the regular expression to identify the WH-orientation of a tweet and categorize the tweet into the *InformationWanted* information type.

A.2.3 Feature Extraction

For our proposed framework, we extract a set of 19 features broadly grouped into 4 different categories, including lexical and content relevance features, incident and event related features, twitter specific features, and sentiment aware features. The feature extraction processes are described in Table A.1.

The first 3 lexical and content relevance features are used to estimate the similarity between a given incident query and a tweet. In this regard, we generate the incident query by combining the query title and narrative and perform the minimal preprocessing as described in Section A.2.1. We also extract 6 incident event related features that seem to be important during the disaster situation. We utilize the Stanford named entity recognizer (NER) tool [201] to extract the location count, organization count, and person count features. Along with this direction, a publicly available library is utilized to estimate the phone number count feature. We also use the CMU ARK POS tagger [202] to identify the noun POS of each tokenized word which is required to extract the noun count feature. To estimate the sentiment polarity of a tweet, we use a publicly available package SentiStrength [181]. We construct the positive and negative sentiment bearing

Table A.1: List of features used in this work.

Feature Type	Feature Name
Lexical and Content Relevance Features	1. TF-IDF [31] similarity score between an incident query and a tweet. 2. Okapi BM25 [34] similarity score between an incident query and a tweet. 3. Language model with Dirichlet smoothing [33] score between an incident query and a tweet. 4. Tweet Length Feature: Number of words available in a tweet. 5. Average Word Length Feature: Average length of the words available in a tweet.
Incident and Event Related Features	1. Location Count Feature: Number of locations name available in a tweet. 2. Organization Count Feature: Number of organizations name available in a tweet. 3. Person Count Feature: Number of person information available in a tweet. 4. Noun Count Feature: Number of noun POS available in a tweet. 5. Phone Number Count Feature: Number of phone number available in a tweet. 6. Known Already Count Feature: Number of previously posted tweets that are closely matched (based on Cosine Similarity) with the corresponding tweet.
Sentiment Aware Features	1. Sentiment Polarity Feature: A binary feature that is assigned to 1 if a tweet has the positive or negative sentiment polarity and 0 otherwise. 2. Positive Word Count Feature: Number of positive words available in a tweet based on the lexicon. 3. Negative Word Count Feature: Number of negative words available in a tweet based on the lexicon. 4. Emoticon Count Feature: Number of emoticons available in a tweet.
Twitter Specific Features	1. Hashtag Feature: A binary feature that is assigned to 1 if a tweet contains a hashtag and 0 otherwise. 2. Hashtag Count Feature: Number of hashtags available in a tweet. 3. URL Feature: A binary feature that is assigned to 1 if a tweet contains a URL and 0 otherwise. 4. Retweet Feature: A binary feature that is assigned to 1 if a tweet is a retweet of other tweet and 0 otherwise.
Total	19 Features

word lexicons as described in [203]. We utilize these lexicons to estimate the lexicon based sentiment aware features. For emoticon count feature, we use a publicly available library to identify the emoticon. Other features are extracted as described in Table A.1.

A.2.4 An Ensemble of Learning Approach

A.2.4.1 Support Vector Machine (SVM) Classifier

We use the $SVM^{multiclass}$ with RBF kernel from [204]. It uses the multi-class formulation described in [205]. For a training set $(x_1, y_1)...(x_n, y_n)$ with labels y_i in $[1..k]$, it finds the solution of the following optimization problem during training:

$$
\begin{aligned}
&min 1/2 \sum_{i=1..k} w_i * w_i + C/n \sum_{i=1..n} \xi_i \\
&\text{s.t. for all y in } [1..k] : \\
&[x_1 * w_{yi}] >= [x_1 * w_y] + 100 * \triangle(y_i, y) - \xi_1 \\
&\ldots\ldots. \\
&\text{s.t. for all y in } [1..k] : \\
&[x_n * w_{yn}] >= [x_n * w_n] + 100 * \triangle(y_n, y) - \xi_n
\end{aligned}
$$

where C is the usual regularization parameter that trades off margin size and training error. We estimate the optimal value of C using cross-validation. $\triangle(y_n, y)$ is the loss function that returns 0 if y_n equals y, and 1 otherwise. To solve this optimization problem, $SVM^{multiclass}$ uses an algorithm based on structural SVMs. For the training, we use the features described in Section A.2.3.

A.2.4.2 Deep Learning based Classifiers

Besides feature based multi-class SVM clasifier, we employ the deep neural network based classifier models because traditional bag-of-words based methods cannot perform well due to the curse of dimensionality and the loss of word order information. However, to train the deep learning models effectively, it is important to represent the tweets as meaningful features. To achieve this goal, we apply the CLSTM architecture inspired by the proposal of Zhou et al. [65].

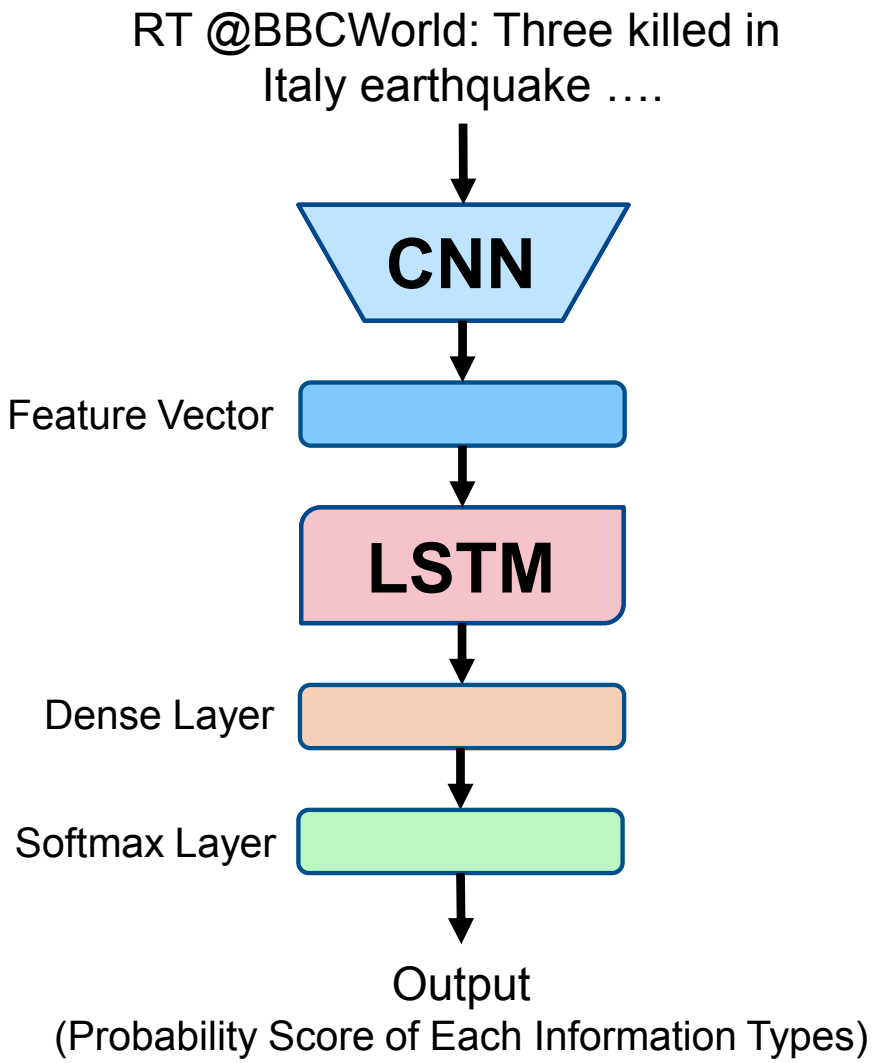

Figure A.2: Convolutional long short-term memory (CLSTM) network.

In our **CLSTM** architecture as depicted in Figure A.2, the higher level representations of CNN are fed into the LSTM to learn long-term dependencies. The CNN is constructed on top of the pre-trained word vectors from *fastText* [54] to learn higher-level representations of n-grams. The feature maps of CNN are then organized as sequential window features to serve as the input of LSTM to learn sequential correlations from higher-level sequence representations. The LSTM

transition functions are defined as follows:

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
u_t &= \phi(W_u \cdot [h_{t-1}, x_t] + b_u) \\
c_t &= f_t \odot c_{t-1} + i_t \odot u_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

where i_t, f_t, o_t, u_t, c_t, and h_t denote the input gate, forget gate, output gate, cell input activation, the cell state, and the current hidden state, respectively, at the current time step t. The symbol σ is the logistic sigmoid function to set the gating values in $[0, 1]$. ϕ is the hyperbolic tangent activation function that has an output in $[1, -1]$ and $\odot$ is the element-wise multiplication.

At the last time step of LSTM, the output of the hidden state is regarded as the final tweet representation and passed to a fully connected softmax layer on top. The output of the softmax layer is the probability distribution over all the information types. To learn the model parameter, we utilize the stochastic gradient descent (SGD) and adopt the Adam optimizer [190].

However, unidirectional LSTM only preserves information of the past context. To understand the context better, bidirectional LSTM is used which runs forward and backward LSTM along with each input sequence and captures both past and future context. The basic idea of bidirectional LSTM is that the output at each time depends on the previous elements and the next elements in the sequence. In a bidirectional LSTM, two LSTMs are stacked on the top of each other. The one that processes the input in its original order and the one that processes the reversed input sequence. The output is then computed based on the hidden state of both LSTMs.

More recently, the attention mechanism has been introduced in the neural network models to mimic human visual attention characteristics that is focus on a certain region of an image and adjusting the focal point over time. Rather than encoding the full source text, the attention mechanism allows the model to learn what to attend based on the input text.

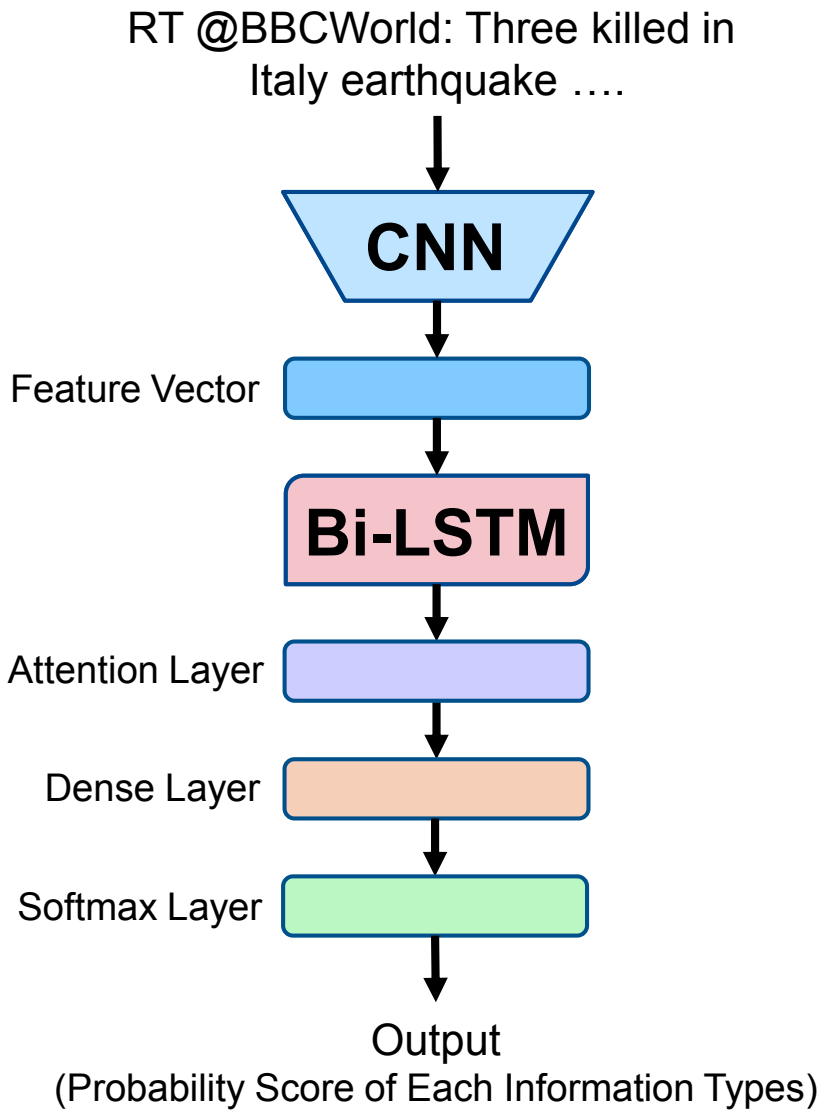

Figure A.3: Attention based convolutional bidirectional LSTM (ACBLSTM) network.

In our **ACBLSTM** architecture as depicted in Figure A.3, the higher level representations of CNN are fed into the bidirectional LSTM to learn long-term dependencies. In order to amplify the contribution of important elements in the final representation of bidirectional LSTM, we applied a recently introduced attention mechanism [206, 207] to aggregate all the hidden states according to their relative importance.

To improve the performance, we utilize the stacked bidirectional LSTM instead of a single bidirectional LSTM in our **ACSBLSTM** architecture. Our stacked architecture is comprised of $N = 15$ bidirectional LSTM layers, where each layer provides a sequence output to the next layer depicted in Figure A.4.

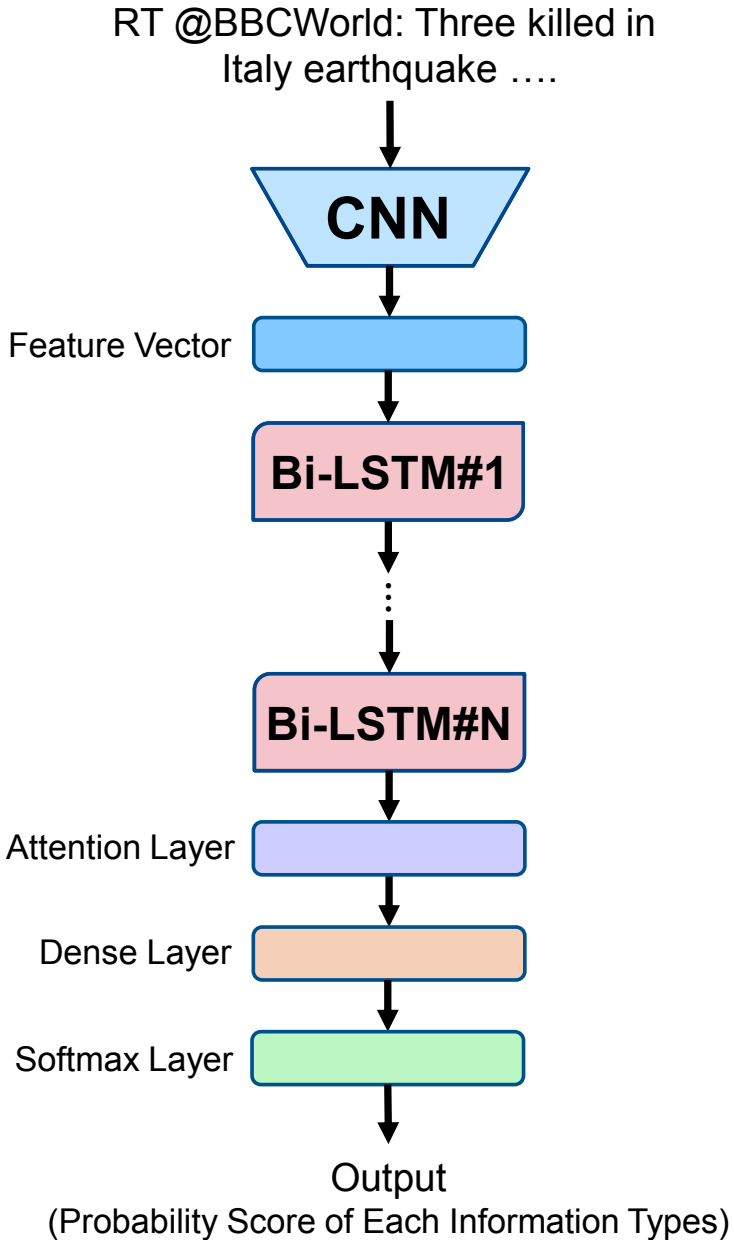

Figure A.4: Attention based convolutional stacked bidirectional LSTM (ACSBLSTM) network.

Next, we employ the state-of-the-art deep learning architecture **DeepMoji (DM)**, proposed by Felbo et al. [208]. We use the DeepMoji architecture without loading the pre-trained weights. As depicted in Figure A.5, DeepMoji uses an embedding layer of 256 dimensions to project each word into a vector space. Two bidirectional LSTM layers with 1024 hidden units in each (512 in each direction) are applied to capture the context of each word. Finally, an attention layer takes all of these layers as input using skip-connections. The representation vector obtained from the attention layer is sent to the softmax layer for classification.

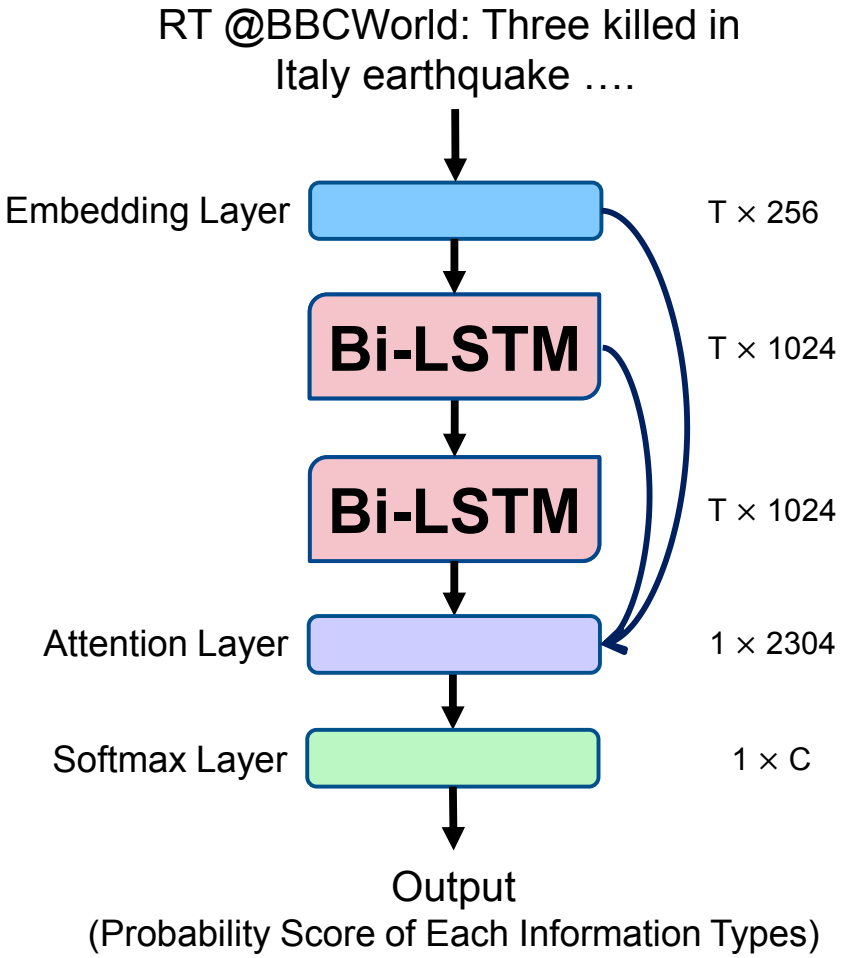

Figure A.5: DeepMoji network, where T is the tweet length and C is the number of classes.

A.2.5 Combining the Classifiers

After developing our proposed rule-based classifier and training the deep neural network based classifiers and support vector machine (SVM) classifier, we combine them to classify the tweet into the high-level information type. At first, our rule-based classifier is applied to classify the tweet to the corresponding information type. Tweets that are not classified by the rule-based classifier we consider the weighted ensemble based prediction from multi-class SVM classifier and several deep neural network models. The prediction score is computed using the following Equation A.2.5.

$$P(C_i|T) = \alpha \cdot P(CL_{DNN}|T) + (1 - \alpha) \cdot P(CL_{SVM}|T)$$

where

$$C_i \in \{\text{List of all high-level information types}\} \text{ and}$$
$$CL_{DNN} \in \{\text{CLSTM, ACBLSTM, ACSBLSTM, DM}\}$$

where, given a tweet T, the final relevance probability score $P(C_i|T)$ is estimated based on the prediction score from a deep neural network model denoted as $P(CL_{DNN}|T)$ and multi-class SVM classifier denoted as $P(CL_{SVM}|T)$. To select the optimal value for the anchoring parameter α, we swept the parameter between $\{0.1, ..., 0.9\}$. Information type that gets the highest relevance probability score will be assigned to the label of the tweet.

A.3 Experiments and Evaluation

A.3.1 Dataset Collection

The TREC incident streams (TREC-IS) task at TREC-2018 provides a benchmark dataset to evaluate the performance of the proposed systems. The dataset contains 21 query topics along with the relevant tweets sampled from several disaster events such as earthquake, typhoon, shooting, etc. Among the 21 query topics, the training set contains 6 query topics and the test set contains 15 query topics. The number of tweets in the training set is around 1300, whereas the number of tweets in the test set is around 20,000. The organizer also provides an

ontology of information types, which contains 25 information types or class label broadly grouped into Request, Report, CallToAction, and Other.

A.3.2 Evaluation Measure

To evaluate the performance of our proposed systems, we applied the evaluation measure used in the TREC-IS task. According to the benchmark of the 2018 TREC-IS task, participant systems were tasked to assign one most representative information type per-tweet. However, during the ground truth generation, the human assessors were allowed to select as many information types as appropriate for a single tweet. Therefore, to evaluate the performance of a TREC-IS system (i.e. how effectively it can categorize the tweets into the 25 high-level information types in the TREC-IS ontology) the organizer used two ways referred to as multi-type and any-type.

In the multi-type evaluation, the categorization performance per information type is estimated in a 1 vs. All manner. If both the system and human assessor selected the corresponding category then the system is considered to correctly categorize a tweet. Whereas, in the any-type evaluation, a system is considered to correctly categorize a tweet if it assigned any of the categories that the human assessor selected for that tweet. Any-type evaluation approach is useful to estimate the overall performance of a 2018 TREC-IS system. Four standard evaluation metrics including precision, recall, F1 score, and accuracy were used in both the multi-type and any-type evaluation criteria.

A.3.3 Results with Different Experimental Settings

We now evaluate the performance of our proposed methods. At first, We describe the experimental settings of each method and the summarized evaluation results for both the multi-type and any-type evaluation criteria were presented in Table A.2.

At first, our rule-based classifier is applied to classify the tweet into corresponding information type and tweets that are not classified by the rule-based classifier, we consider the weighted ensemble based prediction from multi-class SVM classifier and the CLSTM (described in Section A.2.4.2) architecture. The

Table A.2: Performance (Precision, Recall, F1 Score, and Accuracy; higher is better) on TREC-IS 2018 test set for various experimental settings. The best results are highlighted in boldface.

Method	Multi-type (Macro)			
	Precision	Recall	F1 Score	Accuracy
KDEIS1_CLSTM	0.1388	0.0607	0.0620	0.8929
KDEIS2_ACBLSTM	0.1512	0.0689	0.0703	0.8890
KDEIS3_ACSBLSTM	0.1209	0.0577	0.0482	0.8933
KDEIS4_DM	0.1482	0.0708	0.0734	**0.9035**
Participant Median	**0.1827**	**0.0784**	**0.0825**	0.8993

Method	Any-type (Micro)			
	Precision	Recall	F1 Score	Accuracy
KDEIS1_CLSTM	0.2575	0.9783	0.4077	0.2580
KDEIS2_ACBLSTM	0.2089	0.9734	0.3440	0.2098
KDEIS3_ACSBLSTM	0.2630	0.9788	0.4147	0.2635
KDEIS4_DM	0.3914	**0.9856**	**0.5603**	**0.3908**
Participant Median	**0.3978**	0.6164	0.4775	0.3385

prediction score is computed according to Equation A.2.5. We denoted this setting as **KDEIS1_CLSTM**. Next, we used the ACBLSTM deep learning architecture instead of CLSTM in the above setting and referred this setting as **KDEIS2_ACBLSTM**. Along with this direction, we consider the ACSBLSTM and DM based deep neural network architectures in the **KDEIS3_ACSBLSTM** and **KDEIS4_DM** settings, respectively. We also reported the participant median results for comparison.

Experimental results showed that our KDEIS4_DM setting achieved the best performance in both the multi-type and any-type evaluation criteria in terms of primary evaluation measure F1 score. However, for the multi-type evaluation criteria, none of our systems outperform the participant median. Whereas for the any-type evaluation criteria, our KDEIS4_DM system outperformed the participant median by more than 8% in terms of F1 score and by more than 5% in terms of accuracy.

Moreover, we also reported the experimental results of top 5 performing systems in TREC-IS 2018 in Table A.3. It showed that our KDEIS4_DM achieved the second position among the participants.

Table A.3: Top 5 Performing Systems (Precision, Recall, F1 Score, and Accuracy; higher is better) in TREC-IS 2018. Boldfaced one is our proposed system.

Method	Any-type (Micro)			
	Precision	Recall	F1 Score	Accuracy
cbnuS2	0.4559	0.7780	0.5749	0.4213
KDEIS4_DM	0.3914	0.9856	0.5603	0.3908
umdhcilfasttext	0.4534	0.7260	0.5582	0.4022
cbnuS1	0.4472	0.7402	0.5575	0.4064
NHK_run2	0.4483	0.7143	0.5509	0.3997
Participant Median	0.3978	0.6165	0.4775	0.3385

A.4 Discussion

We presented our approach to the TREC 2018 incident streams (TREC-IS) task as described above. We tackled the problem by employing an ensemble of classifiers. Along with a rule-based classifier, four different deep neural network models in combination with a support vector machine is employed in our proposed methods. Among our several experimental settings, KDEIS4_DM achieved the second best performance (F1 Score = 0.5603 for any-type evaluation) in terms of primary evaluation measure.

There is much room left to further improve our methods in TREC-IS task. Shortage of training dataset for our deep learning approach is the main problem. In the future, we have a plan to overcome this limitation by incorporating more training samples collected in an unsupervised manner. We also have a plan to exploit the more sophisticated techniques in our deep learning approaches.

References

[1] Y. Zhang, M. Chen, and L. Liu, "A review on text mining," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2015, pp. 681–685. xi, 13

[2] A.-H. Tan *et al.*, "Text mining: The state of the art and the challenges," in *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, vol. 8. sn, 1999, pp. 65–70. xi, 12, 13

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. xi, 21, 22

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 25, no. 4, pp. 919–931, 2013. 1, 81, 128

[5] M. Basu, S. Bandyopadhyay, and S. Ghosh, "Post disaster situation awareness and decision support through interactive crowdsourcing," *In: Proceedings of International Conference on Humanitarian Technology: Science, Systems and Global Impact (HumTech), Procedia Engineering*, vol. 159, pp. 167–173, 2016. 1

[6] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff, "Overview of the trec-2011 microblog track," in *Proceedings of the 20th Text REtrieval Conference (TREC)*. NIST, 2011. 2, 31, 40, 63, 65, 66, 73, 82, 101, 107, 128

[7] C. Li, Y. Wang, and Q. Mei, "A user-in-the-loop process for investigational search: Foreseer in trec 2013 microblog track." in *TREC*, 2013. 2

[8] J. Teevan, D. Ramage, and M. R. Morris, "# twittersearch: a comparison of microblog search and web search," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 2011, pp. 35–44. 2, 3, 14, 31, 32, 34, 39, 81

[9] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*. IEEE, 2010, pp. 1–10. 4

[10] J. Tang, X. Wang, H. Gao, X. Hu, and H. Liu, "Enriching short text representation in microblog for clustering," *Frontiers of Computer Science*, vol. 6, no. 1, pp. 88–101, 2012. 4

[11] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H.-Y. Shum, "An empirical study on learning to rank of tweets," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*. Association for Computational Linguistics (ACL), 2010, pp. 295–303. 5, 29, 31, 32, 33, 35, 48, 49

[12] D. Metzler and C. Cai, "Usc/isi at trec 2011: Microblog track." in *Proceedings of the 20th Text REtrieval Conference (TREC)*. NIST, 2011. 5, 33, 40, 70, 71, 73, 75

[13] F. Liang, R. Qiang, and J. Yang, "Exploiting real-time information retrieval in the microblogosphere," in *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*. ACM, 2012, pp. 267–276. 5, 32, 33, 40, 70, 71, 73

[14] L. Jia, C. Yu, and W. Meng, "The impacts of structural difference and temporality of tweets on retrieval effectiveness," *ACM Transactions on Information Systems (TOIS)*, vol. 31, no. 4, p. 21, 2013. 5, 32, 70, 71, 72, 73

[15] K. Albishre, Y. Li, and Y. Xu, "Effective pseudo-relevance for microblog retrieval," in *Proceedings of the Australasian Computer Science Week Multiconference (ACSW)*. ACM, 2017, p. 51. 5, 33, 34, 82, 113, 114

[16] T. El-Ganainy, W. Magdy, and A. Rafea, "Hyperlink-extended pseudo relevance feedback for improved microblog retrieval," in *Proceedings of the 1st International Workshop on Social Media Retrieval and Analysis (SoMeRA)*. ACM, 2014, pp. 7–12. 5, 34, 82

[17] M. A. Zingla, L. Chiraz, and Y. Slimani, "Short query expansion for microblog retrieval," *Procedia Computer Science*, vol. 96, pp. 225–234, 2016. 5, 34, 82

[18] J. Rao and J. Lin, "Temporal query expansion using a continuous hidden markov model," in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR)*. ACM, 2016, pp. 295–298. 5, 34, 40, 56, 70, 71, 72, 73, 82, 113, 114

[19] T. Miyanishi, K. Seki, and K. Uehara, "Combining recency and topic-dependent temporal variation for microblog search," in *Advances in Information Retrieval*. Springer, 2013, pp. 331–343. 5, 34, 40, 61, 70, 71, 73, 82, 106, 113

[20] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, "Selecting good expansion terms for pseudo-relevance feedback," in *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2008, pp. 243–250. 5, 35, 80, 82, 91, 92

[21] Z. Zhang, Q. Wang, L. Si, and J. Gao, "Learning for efficient supervised query expansion via two-stage feature selection," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 265–274. 5, 35, 80, 82, 99

[22] A. for Information and I. M. (AIIM), "What is information access?" 2011. 10

[23] M. Hearst *et al.*, "User interfaces and visualization," *Modern Information Retrieval*, pp. 257–323, 1999. 11

[24] C. Manning, P. Raghavan, and H. Sch
"utze, "Introduction to information retrieval," 2008. 11, 63, 64

[25] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2010, pp. 841–842. 13

[26] A. Kyriakopoulou and T. Kalamboukis, "Text classification using clustering," in *The Discovery Challenge Workshop*. Citeseer, p. 28. 13

[27] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137–142, 1998. 13

[28] M. Efron, "Information search and retrieval in microblogs," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 6, pp. 996–1008, 2011. 14, 15, 31

[29] B. Truong, C. Caragea, A. Squicciarini, and A. H. Tapia, "Identifying valuable information from twitter during natural disasters," *Proceedings of the American Society for Information Science and Technology (ASIS&T)*, vol. 51, no. 1, pp. 1–4, 2014. 15, 36

[30] M. Basu, K. Ghosh, S. Das, S. Bandyopadhyay, and S. Ghosh, "Microblog retrieval during disasters: Comparative evaluation of ir methodologies," in *Forum for Information Retrieval Evaluation (FIRE)*. Springer, 2016, pp. 20–38. 15, 27, 36

[31] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge University Press, 2014. 15, 16, 48, 49, 91, 92, 133

[32] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975. 16, 48, 49

[33] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1998, pp. 275–281. 17, 48, 49, 133

[34] S. E. Robertson, S. Walker, M. Beaulieu, and P. Willett, "Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive track," *NIST Special Publication SP*, no. 500, pp. 253–264, 1999. 17, 48, 49, 91, 92, 133

[35] R. L. T. Santos, "Explicit web search result diversification," Ph.D. dissertation, University of Glasgow, 2013. 18, 48, 49

[36] P. Malakasiotis and I. Androutsopoulos, "Learning textual entailment using svms and string similarity measures," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics (ACL), 2007, pp. 42–47. 18, 48, 49

[37] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015. 20

[38] D. Roy, D. Paul, M. Mitra, and U. Garain, "Using word embeddings for automatic query expansion," *arXiv Preprint arXiv:1606.07608*, 2016. 20, 27, 35, 97

[39] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119. 20, 22, 35, 52, 97

[40] S. Kuzi, A. Shtok, and O. Kurland, "Query expansion using word embeddings," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2016, pp. 1929–1932. 20, 27, 35, 82, 113, 114, 115

[41] M. Almasri, C. Berrut, and J.-P. Chevallet, "A comparison of deep learning based query expansion with pseudo-relevance feedback and mutual information," in *European Conference on Information Retrieval (ECIR)*. Springer, 2016, pp. 709–715. 20, 27, 35

[42] F. Diaz, B. Mitra, and N. Craswell, "Query expansion with locally-trained word embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 367–377. 20, 27, 35

[43] H. Zamani and W. B. Croft, "Embedding-based query language models," in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval (ICTIR)*. ACM, 2016, pp. 147–156. 20, 35

[44] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78. 20, 28

[45] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL) (Volume 1: Long Papers)*, vol. 1, 2014, pp. 1555–1565. 20

[46] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011, pp. 142–150. 20

[47] A. N. Chy, M. Z. Ullah, and M. Aono, "Microblog retrieval using ensemble of feature sets through supervised feature selection," *IEICE Transactions on Information and Systems*, vol. 100, no. 4, pp. 793–806, 2017. 20, 34, 35, 82, 99, 113, 114, 115, 117

[48] E. Nalisnick, B. Mitra, N. Craswell, and R. Caruana, "Improving document ranking with dual word embeddings," in *Proceedings of the 25th International Conference Companion on World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, 2016, pp. 83–84. 20, 32, 35

[49] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International Conference on Machine Learning (ICML)*, 2015, pp. 957–966. 20

[50] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, 2013, pp. 746–751. 20

[51] A. Zhila, W.-t. Yih, C. Meek, G. Zweig, and T. Mikolov, "Combining heterogeneous models for measuring relational similarity," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, 2013, pp. 1000–1009. 20

[52] Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014. 20

[53] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. 22, 35

[54] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 22, 135

[55] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2265–2273. 22

[56] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. 22

[57] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014. 23, 28, 35

[58] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015. 23

[59] P. Blunsom, E. Grefenstette, and N. Kalchbrenner, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014. 23

[60] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*. ACM, 2008, pp. 160–167. 23

[61] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012. 23, 25

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986. 23, 25

[63] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 25, 35

[64] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 1041–1044. 25

[65] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *CoRR*, vol. abs/1511.08630, 2015. [Online]. Available: http://arxiv.org/abs/1511.08630 25, 28, 35, 89, 134

[66] A. Ng. (2018) Home - deeplearning.ai. [Online]. Available: https://www.deeplearning.ai/ 27

[67] S.-A. Bahrainian and A. Dengel, "Sentiment analysis using sentiment features," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*. IEEE Computer Society, 2013, pp. 26–29. 27

[68] C. Zhu, X. Qiu, X. Chen, and X. Huang, "A re-ranking model for dependency parser with recursive convolutional neural network," *arXiv preprint arXiv:1505.05667*, 2015. 27

[69] J. Rao, H. He, and J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2016, pp. 1913–1916. 27

[70] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "anmm: Ranking short answer texts with attention-based neural matching model," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2016, pp. 287–296. 27

[71] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft, "Neural ranking models with weak supervision," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 65–74. 27

[72] J. Rao, W. Yang, Y. Zhang, F. Ture, and J. Lin, "Multi-perspective relevance matching with hierarchical convnets for social media search," *arXiv preprint arXiv:1805.08159*, 2018. 27

[73] L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng, "Modeling document novelty with neural tensor network for search result diversification," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 395–404. 27

[74] M. Z. Ullah, M. Shajalal, A. N. Chy, and M. Aono, "Query subtopic mining exploiting word embedding for search result diversification," in *Asia Information Retrieval Symposium (AIRS)*. Springer, 2016, pp. 308–314. 27

[75] M. Z. Ullah and M. Aono, "A bipartite graph-based ranking approach to query subtopics diversification focused on word embedding features," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 12, pp. 3090–3100, 2016. 27

[76] S. Ghosh and K. Ghosh, "Overview of the fire 2016 microblog track: Information extraction from microblogs posted during disasters." in *Working notes of Forum of Information Retrieval (FIRE)*, 2016, pp. 56–61. 27, 36

[77] M. Basu, A. Roy, K. Ghosh, S. Bandyopadhyay, and S. Ghosh, "Microblog retrieval in a disaster situation: A new test collection for evaluation." in *SMERP@ ECIR*, 2017, pp. 22–31. 27, 36

[78] M. Basu, S. Ghosh, K. Ghosh, and M. Choudhury, "Overview of the fire 2017 track: Information retrieval from microblogs during disasters (irmidis)," *Working notes of Forum of Information Retrieval (FIRE)*, pp. 8–10, 2017. 27, 36

[79] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 959–962. 28, 35

[80] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 2539–2544. 28

[81] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," in *26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 3988–3994. 28

[82] Y. Zhou, A. I. Cristea, and L. Shi, "Connecting targets to tweets: Semantic attention-based model for target-specific stance detection," in *International Conference on Web Information Systems Engineering (WISE)*. Springer, 2017, pp. 18–32. 28

[83] P. Wei, W. Mao, and D. Zeng, "A target-guided neural memory model for stance detection in twitter," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8. 28

[84] K. Dey, R. Shrivastava, and S. Kaushik, "Topical stance detection for twitter: A two-phase lstm model using attention," in *European Conference on Information Retrieval*. Springer, 2018, pp. 529–536. 28

[85] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 task 1: Affect in tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval)*, 2018, pp. 1–17. 28

[86] M. Aono and S. Himeno, "Kde-affect at semeval-2018 task 1: Estimation of affects in tweet by using convolutional neural network for n-gram," in *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval)*, 2018, pp. 156–161. 28

[87] B. Kratzwald, S. Ilic, M. Kraus, S. Feuerriegel, and H. Prendinger, "Decision support with text-based emotion recognition: Deep learning for affective computing," *arXiv preprint arXiv:1803.06397*, 2018. 28

[88] H. Naderi, B. H. Soleimani, S. Mohammad, S. Kiritchenko, and S. Matwin, "Deepminer at semeval-2018 task 1: Emotion intensity recognition using deep representation learning," in *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval)*, 2018, pp. 305–312. 28

[89] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 1615–1625. 28

[90] F. Barbieri, L. E. Anke, J. Camacho-Collados, S. Schockaert, and H. Saggion, "Interpretable emoji prediction via label-wise attention lstms," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 4766–4771. 28

[91] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1285–1298. 28

[92] Y. Wang and W. Xu, "Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud," *Decision Support Systems (DSS)*, vol. 105, pp. 87–95, 2018. 28

[93] Y. Song, A. M. Elkahky, and X. He, "Multi-rate deep learning for temporal recommendation," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 909–912. 28

[94] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, and J. Wang, "Dynamic attention deep model for article recommendation by learning human editors' demonstration," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 2051–2059. 28

[95] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017. 28

[96] J. Wehrmann, W. E. Becker, and R. C. Barros, "A multi-task neural network for nultilingual sentiment classification and language detection on twitter," *Machine Translation (MT)*, vol. 2, no. 32, p. 37, 2018. 28

[97] J. C. Chang and C.-C. Lin, "Recurrent-neural-network for language detection on twitter code-switching corpus," *arXiv preprint arXiv:1412.4314*, 2014. 28

[98] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. 28

[99] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014. 28

[100] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016. 28

[101] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *arXiv preprint arXiv:1606.01541*, 2016. 28

[102] P. Liu, X. Zhang, M. Pistoia, Y. Zheng, M. Marques, and L. Zeng, "Automatic text input generation for mobile testing," in *Proceedings of the 39th International Conference on Software Engineering*. IEEE Press, 2017, pp. 643–653. 28

[103] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *arXiv preprint arXiv:1701.06547*, 2017. 28

[104] W. Xi, J. Lind, and E. Brill, "Learning effective ranking functions for newsgroup search," in *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2004, pp. 394–401. 29

[105] K. Fujimura and N. Tanimoto, "The eigenrumor algorithm for calculating contributions in cyberspace communities," in *Trusting Agents for Trusting Electronic Societies*. Springer, 2005, pp. 59–74. 29

[106] S.-K. Han, D. Shin, J.-Y. Jung, and J. Park, "Exploring the relationship between keywords and feed elements in blog post search," *World Wide Web*, vol. 12, no. 4, pp. 381–398, 2009. 29

[107] A. Kritikopoulos, M. Sideri, and I. Varlamis, "Blogrank: Ranking weblogs based on connectivity and similarity features," in *Proceedings of the 2nd International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications*. ACM, 2006, p. 8. 30

[108] G. Xu and W.-Y. Ma, "Building implicit links from content for forum search," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2006, pp. 300–307. 30

[109] H. Liu, J. Yang, J. Wang, and Y. Zhang, "A link-based rank of postings in newsgroup," in *Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 392–403. 30

[110] Z. Chen, L. Zhang, and W. Wang, "Postingrank: Bringing order to web forum postings," in *Information Retrieval Technology*. Springer, 2008, pp. 377–384. 30

[111] Y. Chen, F. S. Tsai, and K. L. Chan, "Machine learning techniques for business blog search and mining," *Expert Systems with Applications*, vol. 35, no. 3, pp. 581–590, 2008. 30

[112] M. Joshi and N. Belsare, "Blogharvest: Blog mining and search framework." in *In Proceedings of the International Conference on Management of Data (COMAD)*. Citeseer, 2006, pp. 226–229. 30

[113] H. Kuwata, M. Oka, and H. Mori, "Searching blog sites with product reviews," in *International Conference on Human Interface and the Management of Information*. Springer, 2013, pp. 495–500. 30

[114] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin, "Overview of the trec-2012 microblog track," in *Proceedings of the 21st Text REtrieval Conference (TREC)*, 2012, p. 20. 31, 101

[115] J. Lin and M. Efron, "Overview of the trec-2013 microblog track," in *Proceedings of the 22nd Text REtrieval Conference (TREC)*. NIST, 2013. 31, 58

[116] J. Lin, M. Efron, Y. Wang, and G. Sherman, "Overview of the trec-2014 microblog track," 2014. 31, 38

[117] Y. Kim, R. Yeniterzi, and J. Callan, "Overcoming vocabulary limitations in twitter microblogs," in *Proceeddings of the 21st Text REtrieval Conference (TREC)*. NIST, 2012. 32

[118] N. Kanhabua, R. Blanco, K. Nørvåg *et al.*, *Temporal Information Retrieval*. now Publishers, 2015. 32

[119] M. Efron, J. Lin, J. He, and A. de Vries, "Temporal feedback for tweet search with non-parametric density estimation," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2014, pp. 33–42. 32, 34

[120] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *Advances in Information Retrieval*. Springer, 2011, pp. 362–367. 32, 33, 34

[121] M. Efron and G. Golovchinsky, "Estimation methods for ranking recent information," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2011, pp. 495–504. 32

[122] T. Miyanishi, K. Seki, and K. Uehara, "Improving pseudo-relevance feedback via tweet selection," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2013, pp. 439–448. 32, 33, 34

[123] R. Aly, T. Demeester, and S. Robertson, "Probabilistic models in ir and their relationships," *Information Retrieval*, vol. 17, no. 2, pp. 177–201, 2014. 32

[124] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones, "Word embedding based generalized language model for information retrieval," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 795–798. 32, 35

[125] F. Damak, K. Pinel-Sauvagnat, M. Boughanem, and G. Cabanac, "Effectiveness of state-of-the-art features for microblog search," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC)*. ACM, 2013, pp. 914–919. 32

[126] A. Severyn, A. Moschitti, M. Tsagkias, R. Berendsen, and M. De Rijke, "A syntax-aware re-ranker for microblog retrieval," in *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2014, pp. 1067–1070. 32

[127] J. A. R. Perez, A. J. McMinn, and J. M. Jose, "University of glasgow (uog_twteam) at trec microblog," in *Proceedings of the 21st Text REtrieval Conference (TREC)*. NIST, 2012. 33

[128] G. Amodeo, G. Amati, and G. Gambosi, "On relevance, time and query expansion," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2011, pp. 1973–1976. 33, 55

[129] J. Choi, W. B. Croft, and J. Y. Kim, "Quality models for microblog retrieval," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2012, pp. 1834–1838. 33

[130] F. Fan, R. Qiang, C. Lv, and J. Yang, "Improving microblog retrieval with feedback entity model," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2015, pp. 573–582. 33, 35, 70, 71, 72

[131] J. A. Rodriguez Perez and J. M. Jose, "On microblog dimensionality and informativeness: Exploiting microblogs' structure and dimensions for ad-hoc retrieval," in *Proceedings of the 2015 ACM International Conference on the Theory of Information Retrieval (ICTIR)*. ACM, 2015, pp. 211–220. 33

[132] A. N. Chy, M. Z. Ullah, and M. Aono, "Combining temporal and content aware features for microblog retrieval," in *Proceedings of the 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*. IEEE, 2015, pp. 1–6. 33

[133] Z. Wang and M. Zhang, "Feedback model for microblog retrieval," in *International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 2015, pp. 529–544. 34

[134] M. Roick, M. Jenders, and R. Krestel, "How to stay up-to-date on twitter with general keywords," in *LWA 2015 at CEUR Workshop Proceedings*, 2015, pp. 373–381. 34

[135] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003. 34, 86

[136] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296. 34, 86

[137] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, p. 391, 1990. 34, 113, 114, 115

[138] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. ACM, 2013, pp. 1445–1456. 34, 86, 114

[139] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2010, pp. 787–788. 35

[140] I. Vulić and M.-F. Moens, "Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 363–372. 35

[141] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1422–1432. 35

[142] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428–2437. 35

[143] C. Xiong and J. Callan, "Query expansion with freebase," in *Proceedings of the 2015 ACM International Conference on the Theory of Information Retrieval (ICTIR)*. ACM, 2015, pp. 111–120. 35

[144] B. Xu, H. Lin, and Y. Lin, "Assessment of learning to rank methods for query expansion," *Journal of the Association for Information Science and Technology (JASIST)*, vol. 67, no. 6, pp. 1345–1357, 2016. 35

[145] R. McCreadie, C. Buntain, and I. Soboroff. (2018) Guidelines v1.0 - trec 2018 incident streams track. 36

[146] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh, "Extracting situational information from microblogs during disaster events: A classification-summarization approach," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 583–592. 36

[147] R. Dutt, K. Hiware, A. Ghosh, and R. Bhaskaran, "Savitr: A system for real-time location extraction from microblogs during emergencies," in *Proceedings of the 2018 International Conference Companion on World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, 2018, pp. 1643–1649. 36

[148] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, "Collaborative personalized tweet recommendation," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2012, pp. 661–670. 37

[149] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, and W. Nejdl, "Real-time top-n recommendation in social streams," in *Proceedings of the 6th ACM Conference on Recommender Systems*. ACM, 2012, pp. 59–66. 37

[150] P. Bedi, S. K. Agarwal, P. Vashisth, S. Sharma, and T. Aggarwal, "Online tweet recommendation using extreme learning machine," in *Proceedings of the 2014 Recommender Systems Challenge*. ACM, 2014, p. 62. 37

[151] J. Yu, Y. Shen, and Z. Yang, "Topic-stg: Extending the session-based temporal graph approach for personalized tweet recommendation," in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2014, pp. 413–414. 37

[152] B. Liu and U. Hengartner, "ptwitterrec: a privacy-preserving personalized tweet recommendation framework," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security.* ACM, 2014, pp. 365–376. 37

[153] R. Harakawa, D. Takehara, T. Ogawa, and M. Haseyama, "Sentiment-aware personalized tweet recommendation through multimodal ffm," *Multimedia Tools and Applications (MTA)*, pp. 1–19, 2018. 37

[154] D. P. Karidi, Y. Stavrakas, and Y. Vassiliou, "Tweet and followee personalized recommendations based on knowledge graphs," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 2035–2049, 2018. 37

[155] X. Zeng, J. Li, L. Wang, N. Beauchamp, S. Shugars, and K.-F. Wong, "Microblog conversation recommendation via joint modeling of topics and discourse," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, 2018, pp. 375–385. 37

[156] H. Huang, Q. Zhang, X. Huang *et al.*, "Mention recommendation for twitter with end-to-end memory network," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 17, 2017, pp. 1872–1878. 37

[157] W. Xu, R. Grishman, A. Meyers, and A. Ritter, "A preliminary study of tweet summarization using information extraction," *NAACL 2013*, p. 20, 2013. 38

[158] Z. Ren, S. Liang, E. Meij, and M. de Rijke, "Personalized time-aware tweets summarization," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 2013, pp. 513–522. 38

[159] W. Li, C. Eickhoff, and A. P. de Vries, "Interactive summarization of social media," in *Proceedings of the 5th Information Interaction in Context Symposium*. ACM, 2014, pp. 312–315. 38

[160] A. Zubiaga, D. Spina, E. Amigó, and J. Gonzalo, "Towards real-time summarization of scheduled events from twitter streams," in *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*. ACM, 2012, pp. 319–320. 38

[161] L. Shou, Z. Wang, K. Chen, and G. Chen, "Sumblr: Continuous summarization of evolving tweet streams," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2013, pp. 533–542. 38

[162] V. Rakesh, C. K. Reddy, D. Singh, and M. Ramachandran, "Location-specific tweet detection and topic summarization in twitter," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 1441–1444. 38

[163] J. Lin, A. Roegiest, L. Tan, R. McCreadie, E. Voorhees, and F. Diaz, "Overview of the trec 2016 real-time summarization track," in *Proceedings of the 25th Text REtrieval Conference (TREC)*, vol. 16, 2016. 38

[164] G. Amati, G. Amodeo, M. Bianchi, G. Marcone, F. U. Bordoni, C. Gaibisso, G. Gambosi, A. Celi, C. Di Nicola, and M. Flammini, "Fub, iasi-cnr, univaq at trec 2011 microblog track." in *Proceedings of the 20th Text REtrieval Conference (TREC)*, 2011. 40, 70, 71, 73, 76

[165] F. R. Q. Y. Liang and Y. F. J. Yang, "Pkuicst at trec 2012 microblog track," in *Proceedings of the 21st Text REtrieval Conference (TREC)*, vol. 12. NIST, 2012, p. 19. 40, 70, 72, 73

[166] M. E. Renda and U. Straccia, "Web metasearch: rank vs. score based rank aggregation methods," in *Proceedings of the 2003 ACM Symposium on Applied Computing*. ACM, 2003, pp. 841–846. 42, 85

[167] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics (ACL), 2012, pp. 421–432. 42, 103, 131

[168] F. Liu, F. Weng, and X. Jiang, "A broad-coverage normalization system for social media language," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL): Long Papers-Volume 1*. Association for Computational Linguistics (ACL), 2012, pp. 1035–1044. 42, 103, 131

[169] R. Nagmoti, A. Teredesai, and M. De Cock, "Ranking approaches for microblog search," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1. IEEE, 2010, pp. 153–157. 49, 50

[170] L. Zhao, Y. Zeng, and N. Zhong, "A weighted multi-factor algorithm for microblog search," in *International Conference on Active Media Technology (AMT)*. Springer, 2011, pp. 153–161. 49, 50

[171] T. El-Ganainy, Z. Wei, W. Magdy, and W. Gao, "Qcri at trec 2013 microblog track," in *Proceeddings of the 22nd Text REtrieval Conference (TREC)*. NIST, 2013. 49, 50, 51

[172] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26. 52

[173] A. Gelman, "Bursts: The hidden pattern behind everything we do," *Physics Today*, vol. 63, no. 5, pp. 46–46, 2010. 55

[174] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003. 55

[175] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. 57, 99

[176] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2006, pp. 217–226. 57, 61, 66

[177] G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009, pp. 758–759. 57, 63, 100, 107

[178] J. Friedman, T. Hastie, and R. Tibshirani, "glmnet: Lasso and elastic-net regularized generalized linear models," *R Package Version*, vol. 1, 2009. 59, 103

[179] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: http://CRAN.R-project.org/doc/Rnews/ 60, 103, 105

[180] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 431–439. 60, 100, 103, 105

[181] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 63, no. 1, pp. 163–173, 2012. 63, 106, 132

[182] E. M. Voorhees and D. Harman, "Common evaluation measures," in *The 12th Text REtrieval Conference (TREC 2003)*, 2006, pp. 500–255. 63

[183] C. Buckley and E. M. Voorhees, "Evaluating evaluation measure stability," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM, 2017, pp. 235–242. 64

[184] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002. 65

[185] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 89–96. 65

[186] Z. Han, X. Li, M. Yang, H. Qi, S. Li, T. Zhao, Z. Han, and H. Qi, "Hit at trec 2012 microblog track," in *Proceedings of the 21st Text REtrieval Conference (TREC)*, vol. 12. NIST, 2012, p. 19. 70, 72, 73

[187] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2001, pp. 120–127. 82, 113, 114, 115, 117

[188] J. J. Rocchio, "Relevance feedback in information retrieval," *The SMART retrieval system: Experiments in automatic document processing*, pp. 313–323, 1971. 82, 113, 114, 115, 117

[189] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004. 85

[190] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 90, 136

[191] J. A. R. Perez, A. J. McMinn, and J. M. Jose, "University of glasgow (uog_twteam) at trec microblog 2013." in *Proceedings of the 22nd Text REtrieval Conference (TREC)*. NIST, 2013. 91, 92

[192] J. W. Reed, Y. Jiao, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson, "Tf-icf: A new term weighting scheme for clustering dynamic data streams," in *5th International Conference on Machine Learning and Applications, 2006 (ICMLA'06).s*. IEEE, 2006, pp. 258–263. 91, 92

[193] H. J. Peat and P. Willett, "The limitations of term co-occurrence data for query expansion in document retrieval systems," *Journal of the American Society for Information Science*, vol. 42, no. 5, p. 378, 1991. 91

[194] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annuals of Statistics*, pp. 2769–2794, 2007. 95

[195] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: New features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012. 106

[196] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. 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*. Association for Computational Linguistics, 2011, pp. 42–47. 106

[197] M. W. Berry, S. T. Dumais, and G. W. OBrien, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, no. 4, pp. 573–595, 1995. 115

[198] Y. Lv and C. Zhai, "A comparative study of methods for estimating query language models with pseudo feedback," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*. ACM, 2009, pp. 1895–1898. 115

[199] U. A. Siddiqua, A. N. Chy, and M. Aono, "Stance detection on microblog focusing on syntactic tree representation," in *International Conference on Data Mining and Big Data (DMBD)*. Springer, 2018, pp. 478–490. 130

[200] (2018) Java Tools - language-detection. [Online]. Available: https://code.google.com/p/language-detection/ 131

[201] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2005, pp. 363–370. 132

[202] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, "Improved part-of-speech tagging for online conversational text with word clusters." Association for Computational Linguistics (ACL), 2013. 132

[203] U. A. Siddiqua, T. Ahsan, and A. N. Chy, "Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog," in *19th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2016, pp. 304–309. 134

[204] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the 21st International Conference on Machine Learning (ICML)*. ACM, 2004, p. 104. 134

[205] K. Krammer and Y. Singer, "On the algorithmic implementation of multiclass svms," *Proceedings of JMLR*, pp. pages 265–292, 2001. 134

[206] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015. 137

[207] N. Wang, J. Wang, and X. Zhang, "Ynu-hpcc at ijcnlp-2017 task 4: Attention-based bi-directional gru model for customer feedback analysis task of english," *Proceedings of the IJCNLP 2017, Shared Tasks*, pp. 174–179, 2017. 137

[208] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 139