A UNIFIED FRAMEWORK FOR EVENT RELATED INFORMATION SEEKING

by

Kuang Lu

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering

Fall 2020

A UNIFIED FRAMEWORK FOR EVENT RELATED INFORMATION SEEKING

by

Kuang Lu

Approved: ______________________________

Jamie D. Phillips, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: ______________________________

Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: ______________________________

Louis F. Rossi, Ph.D.
Vice Provost for Graduate & Professional Education and Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Hui Fang, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Austin Brockmeier, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Haining Wang, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ____
Vijay Shanker, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my advisor Prof. Hui Fang, for her continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this dissertation. Without her guidance and persistent help, this dissertation would not have been possible.

Moreover, I would like to thank my dissertation committee members, Prof. Austin Brockmeier, Prof. Haining Wang, and Prof. Vijay Shanker. Thank all of you for the valuable comments and suggestions, as well as hard questions.

I would also like to thank my colleagues and collaborators Monica Rodriguez Guevara, Xitong Liu, Deigo Roa, Hao Wu, Peilin Yang, Yue Wang, Wei Zhong, Hao Xu, Ye Wang, Zitong Cheng, Di Zhao, Disen Wang, Haibi Hu, Yue Zhang, Victor Yang, and Prof. Jimmy Lin for the stimulating discussions and tremendous help.

Last but certainly not the least. I want to thank my parents for their unconditional love and support throughout my life, especially lately during the pandemic.

TABLE OF CONTENTS

Appendix

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Events of different types and scopes are happening every day and influencing nearly every aspect of people's life. In order to obtain information about the events to understand them, search is usually the tool to use. Therefore, it is crucial to provide adequate support for event related information. News and microblogs are two popular information sources to be searched. In this work, we focus on providing effective search techniques for event related queries. The challenges of event related information seeking on these two sources, as well as the inadequacy of the existing methods in terms of overcoming the challenges, are discussed. Based on this, we propose a unified framework to assist event related search interests on news and microblogs that not only includes novel techniques to address the challenges, but also bridge the retrieval on them to better satisfy the search interests, meaning to leverage background information from news to refine the results on microblogs.

On news retrieval, we argue the importance of background information and that it is essential to provide dedicated background information retrieval support. Through analyzing background information from temporal and semantic perspective, a time filter and an aspect based background retrieval model are proposed. Experiments on the TREC News Track data set illustrate that these two methods can work in concert to provide statistically significant improvements over a competitive baseline. Further analysis demonstrates the usefulness of the individual components of the model, such as aspect identification and aspect language model estimation.

On microblog retrieval, due to the lack of long term search interest support and the unsatisfactory pseudo relevance feedback performance, a novel relevance signal called *query collectivity* is proposed, which measures relevance by using the collective presence of multiple query terms. For long term search interests, this measure is used to detect the time periods (e.g. days) when there is no relevant information and informs the system to not return any results. However, when the measure is employed for pseudo relevance feedback, expansion terms are selected in a conservative fashion, and the measure is able to only select them for queries when there is a potential of improving the retrieve effectiveness. Experiment results on TREC Microblog data sets show that the measure achieves advantageous performances on both tasks.

Besides concentrating on enhancing the retrieval for different genres individually for event related searches, we also propose to bridge the two types of retrieval by leveraging the background aspects from news to re-rank the initial retrieval results on microblogs on a per aspect basis to provide an option of exploring microblogs in finer granularity. We carefully design an experiment and its results suggest that the news aspects can represent meaningful event related search interests and, with the help of language models of the aspects that can be obtained from the background news retrieval, initial microblog results can be effectively re-ranked for the aspects.

Chapter 1

INTRODUCTION

Different events, small and big, are happening around the world every day, ranging from the opening of a restaurant chain at a new location to scandals of a national political figure. A person needs to be well informed about some of the events that are related to him or her in order to make important decisions, such as picking a place for the next family gathering or voting in the next national election. To satisfy event related information needs, users usually formulate the needs as queries and search them on some information sources. Arguably, two of the most important and frequently used ones are *news* and *microblogs*. On one hand, news offers well written and well researched articles about the event(s) that can provide basic information about an event, such as the "five w's" (who, what, when, where, why), to help readers know what happened, as well as contextual and background information that can help readers better understand the events. Microblogs, on the other hand, can provide similar information but in a more timely fashion due to its immediacy nature. Besides, if an event is an ongoing or breaking event, compared with news, microblog is a more suitable information source to monitor the development of the event as well as observe public opinions about the event.

However, there are challenges, either in general or event-specific, that might not be addressed adequately by existing techniques. On the news side, there is a lack of support for retrieving *background information*, which is an important type of information need for events. On the microblog side, relatively long term interests are not intensively studied in the literature and classic ad-hoc retrieval techniques might not be sufficient. Moreover, some of the existing techniques, such as pseudo relevance feedback, do not seem to enjoy similar success that they have on document retrieval [65, 65].

Besides these challenges, we also argue that there are potential benefits of bridging news and microblog retrieval for event related queries. Not only can searches on news help the retrieval on microblogs, but also the relevant information on these two sources can complement each other. As a result, in this work we design a *unified* system which consists of a news retrieval pipeline and a microblog retrieval pipeline to handle the event related information seeking on these two sources. In the system, we propose novel techniques to solve the challenges mentioned above, and to bridge the event related retrieval on the two sources.

1.1 Background Information Retrieval for Events on News

On the news side, we argue that ad-hoc retrieval methods might not be sufficient for event related search interests since they lack the support for retrieving *background information*. Fox, a professional journalist and editor, wrote in his book of news writing that background information of an event is the "factual information that a reader needs to fully understand a story" [18]. He further explains that it can be as simple as basic information of the event, such as personal information of the people involved. It also can be *connections* between the reported event and other related events. Background information may certainly be important to be provided alongside the information of the event since it could help explain the history behind the event or some long-term trends that the event is involved in. For instance, given an event about a new development of a major criminal case, readers may certainly want to know previous developments and the origin of the case, without which the story about the new development might be meaningless. Another example is that, given the news event of a wildfire which scorched more than 7,000 acres in September 2020 that was caused by a gender reveal party [1], readers may also be interested in a story that such a party accidentally caused

[1] https://cnn.it/3c3vZKg

a devastating wildfire two years ago [2]. These two stories together can help readers to understand the danger of using fireworks in a dry forested area during a gender reveal party.

Despite the importance of background information, the amount of it provided by merely using traditional ad-hoc retrieval techniques might not be sufficient for the readers. If an event related query is searched using some classic ad-hoc retrieval methods, it is likely that articles closely related to the event, such as the reporting of the event or opinion pieces about the event, are returned. Although they usually contain background information, due to the fact that the focus of them is not to explain clearly the background, there may be an only limited amount of background information. For instance, in the criminal case example above, a reader may still need to read the initial reporting of the case in order to obtain enough details to fully understand the origin of the case. Moreover, in order to not break the flow of the main story, a technique for mentioning background information called information weaving is often used in news writing, which introduces background information in subordinate clauses, while describing the main story in the main clause [19]. It is clear that if a piece of background information is mentioned in this way, the details of it are limited.

Thus, in the news retrieval pipeline of the proposed unified framework, we focus on using the mentioning of the background information in the articles reporting the event to retrieve more details of background information. This is accomplished by introducing an *event background retrieval* step following an initial ad-hoc news retrieval step. First, the articles reporting or discussing events are identified from the initial results through a rule based method to extract the mentioning of background information from. For clarity, these articles are called *event articles* and the results of the second background retrieval step are called *background articles*. As suggested by the definition, background information is multi-faceted. Therefore, we propose an *aspect based background retrieval framework* which identifies the *aspects* of background

[2] https://cnn.it/2RtI3Lh

in event articles, retrieve news articles of background information for the aspects, and estimate the importance of the aspects so the results of different aspects can be merged. The retrieval part is handled by existing retrieval techniques and we focus on aspect identification and aspect importance estimation. For aspect identification, we propose an entity graph based method motivated by our intuitions that first, the usage of different entities indicates the mentioning of different aspects; second, frequently co-occurring entities are likely to belong to the same aspect. More specifically, an entity graph is built given an event article to represent the co-occurrence information of entities among the paragraphs in the article. The graph is segmented using Louvain method [6] so that entities closely connected to each other (i.e. co-occur frequently with each other) are grouped together and deemed as belonging to the same aspects. Aspect labels are then assigned to paragraphs based on the entities in the paragraphs and which aspects these entities are members of. In order to retrieve background information for the aspects, the language models of the aspects then are estimated from the paragraphs with corresponding aspect labels using either L-LDA [41], which is an existing method that leverages labels for the latent Dirichlet allocation (LDA) process, or a proposed method L-PLSA, which is inspired by L-LDA and is a modified version of PLSA [29, 21]. Regarding aspect importance, we propose two ways of measuring them, which are based on how closely they are related to the event, and how unclear or lack of details their descriptions are in the event articles. Background articles of different aspects are then merged based on their importance.

Besides the aspect based background retrieval framework, the temporal characteristics of background information are also leveraged to further improve the effectiveness of background information retrieval. We argue that, intuitively, background information is more likely to be information about events that happened prior to the event that is being searched. Based on this intuition, when retrieving background articles given an event article, a simple time filter is proposed to filter out articles published after the publication of the event article.

The rule-based event article identification method as well as the two background retrieval methods are then tested and analyzed on the background linking task of the 2018 and 2019 News Track, which are designed for background retrieval. Experiment results show the rule based method can identify event articles with high precision and recall. Moreover, both background retrieval methods seem to be useful for background retrieval. Not only does our intuition about the background information being historic seem to be true, but also the simple time filter method can improve the effectiveness of background information retrieval against a competitive baseline. In addition, for the aspect based background retrieval framework, it is also observed that, if appropriate aspect importance estimation methods and aspect language model estimation methods are chosen, the performance can be further improved. Component analyses indicate that the entity graph based aspect identification method helps confine the aspect language model estimation to the paragraphs that describe the corresponding aspects, which, in turn, offers more accurate estimates of the aspect language models.

1.2 Long Term Search Interest Support and Pseudo Relevance Feedback Improvement on Microblogs

In addition to news, microblog is another information source that can provide valuable information if a user is interested in a news event. However, classic ad-hoc retrieval might not be optimal since it lacks the support for long term search interests regardless of whether the query is event related or not. Due to the nature of the topics that a user is interested in, some of them are only actively discussed on microblogs for a short period of time, an example of which is a sports team winning the championship of a tournament. For these topics, one-time ad-hoc searches might be sufficient. However, some topics are expected to have relatively long-lasting discussions. An example is the death of an American black man George Floyd under police custody [3]. The discussion on microblogs around it lasted for a sustained period of time instead of in a short spike. According to a report from the Pew Research Center [39], after the video of the death of George Floyd surfaced, the usage of the hashtag “#BlackLivesMatter”,

[3] https://en.wikipedia.org/wiki/Killing_of_George_Floyd

which is often used alongside discussion of the death of African Americans caused by police brutality, soared on Twitter and the high level of usage sustained for more than 10 days. Therefore, in order to keep up with the newly available information for such a topic, if the classic ad-hoc retrieval paradigm is adopted, a user then needs to search the same query periodically with a relatively short time window (e.g. a day). It might not be a reasonable practice and seems unnecessary to request a user to perform the same action periodically. Thus, besides ad-hoc search, it may be desirable for a system to provide an option to automatically perform a search periodically for the users and daily search seems to be a reasonable frequency.

However, although many topics can have long-lasting popularity on microblogs, not all of them are as significant as the death of George Floyd. As a result, there might not always be new relevant information for them every day for a sustained period of time. For instance, if a user wants to follow a sports team during a tournament, the team might be intensively discussed during the days when they play. However, when they do not, the discussions may shift to the teams that do play. This phenomenon of the lack of relevant information for a certain period of time on microblogs is first discovered in TREC 2015 Microblog Track and a day with no relevant information is called a *silent day* [25]. If a system performs a search in a silent day and returns the irrelevant results to the users, it may waste the time and effort of the users to read the irrelevant results. This, in turn, may hurt the user experience and confidence of the system. Thus, it would be necessary to *detect silent days* and return no results for these days. Silent day detection is an essential part of the microblog retrieval pipeline of the proposed framework. We envision that in practice, users are given the option of searching a query one time or once per day until it is canceled. If the latter choice is chosen, the system will perform silent day detection and return results when the day is non-silent. It is important to note that although we design the system using daily search as the frequency, different frequencies such as hourly or monthly searches can be easily adopted.

Although the above system is able to provide periodical searches for relatively long term interests which complements ad-hoc retrieval, there is another drawback of classic ad-hoc retrieval methods for microblog retrieval that it cannot address. We argue that classic *pseudo relevance feedback* methods are not as effective on microblog retrieval as they are on document retrieval, which might be a result of the ineffectiveness of the *classic retrieval signals* on microblog retrieval. Unlike document retrieval, microblog retrieval mainly deals with shorter text. As a result, retrieval signals such as term frequency (TF) and document length are not very effective [56]. Term frequencies are usually one for all terms, and the variance of the lengths of microblog posts might be insignificant. Inverse document frequency (IDF), on the other hand, becomes the dominant factor for ranking microblog posts. This can be problematic especially under pseudo relevance feedback settings. Pseudo relevance feedback methods, such as RM3 [1], are widely used as a means to improve the performance on basic retrieval functions. Generally, they use basic retrieval functions to perform a round of retrieval. Expansion terms from the top documents of the results are combined with original query terms to perform another round of retrieval. It is clear that the quality of the expansion terms is essential to the effectiveness of the second round of retrieval. The scores of the documents in the first round correlate to the influence of them in selecting expansion terms. Due to the dominant impact of IDF on microblogs, an irrelevant microblog post containing a single high IDF query term can have a score only slightly lower than a relevant one containing the same term as well as other low IDF query terms. As a result, the difference in their impact on selecting expansion terms might be smaller than ideal. This may in turn lead to similar contributions of these two posts to the expansion terms, causing irrelevant terms from the irrelevant post to be selected. Thus, there may be a need for additional retrieval signals that can better differentiate irrelevant microblog posts from relevant ones. Based on the above example, signals favoring posts that cover multiple query terms might satisfy this need.

As a result, we propose a new type of retrieval signals, i.e, *query collectivity*, which can be used to model the relevance based on the collective presence of multiple query terms. Specifically, we propose a general scoring function for query collectivity and instantiate two types of measures based on the function. The first type of the measures is called *phrased-based weighted information gain* (PWIG), which is inspired by the weighted information gain [62]. It is designed to infer relevance by only using the sum of the information gain of the appearances of groups of query terms instead of that of single terms. The second type of the measures is called *local query term coherence* (LQC). It evaluates the degree of query collectivity by computing how often the query terms collectively occur. These two types of signals can be used for not only pseudo relevance feedback, but also silent day detection since the latter also requires the assessment of relevance. Therefore, on the microblog side of the framework, we focus on investigating the usefulness of the proposed signals on silent day detection and pseudo relevance feedback.

For silent day detection, the task is naturally formulated as a classification problem. The proposed measures PWIG and LQC are used as features and compared with features based on state-of-the-art query performance predictors. Besides, we introduced another baseline that was proposed for missing content detection, a similar problem to silent day detection but concerning documents. Experiments are conducted on multiple TREC microblog collections [36, 25, 26, 24], and results show that using the proposed measures alone can outperform both sets of baselines.

For pseudo relevance feedback, the proposed measures are used to directly select expansion terms in a very conservative way. A non-query term is only selected if, by adding it to the original query, the proposed measures for the expanded query, such as PWIG and LQC, increase compared with that of the original query. It might be argued that terms chosen in such a more restricted way are more likely to be related to the original query. The pseudo relevance feedback experiments are conducted on the same sets of microblog data, and statistically significant improvements over traditional pseudo relevance feedback baseline is observed when the proposed method is able to select expansion terms.

1.3 Bridging News and Microblog Retrieval for Event Related Search

In addition to proposing novel techniques dedicated to different information sources, we also try to leverage the information from the news to bridge the two types of information for search interests pertaining to events. More specifically, we propose to use the aspects inferred from news articles to perform retrieval on microblogs. Information from news is chosen since news articles are generally of higher quality and more informative compared to microblog posts. Besides, for an event related query searched on microblogs, there can be multiple aspects of the underlying information need, similar to the background of an event. More importantly, we also argue that some of the more important aspects for an event query on microblogs overlap with the background aspects which can be inferred from the event articles. For instance, in the reporting of the death of George Floyd, the killing of an African American woman Breonna Taylor by police officers around two weeks before is often mentioned [4]. If a reader wants information about George Floyd's death, he or she might also be interested in learning about the shootings of Breonna Taylor. Thus, for event related queries, aspects from news could provide finer granularity for searches on microblogs if the retrieved microblog posts can be re-ranked and re-organized for individual aspects. Moreover, microblog posts are more likely to be opinionated, which is complementary to the factual background information from the news. In order to test the usefulness of the aspects mined from news articles on microblogs, we design an experiment that first ranks the aspects from multiple top documents from the background retrieval step of the news pipeline with the aspect weights, which are estimated by the proposed aspect importance estimation method, as well as the document relevance scores, which are produced by the first news retrieval step. Top aspects then are used to retrieve microblog posts individually. The average performance of the retrieval results for top aspects shows statistically significant improvement over that of a baseline that retrieves microblog posts for each of the top documents. This seems to indicate that the top

[4] https://bit.ly/3cTiIVe

aspects from news articles can represent different and more specific information needs on microblogs.

The following chapters of the thesis are organized as follows: the existing work to the proposed framework is discussed in chapter 2. It is followed by chapter 3 in which the overall design of the framework is described. In chapter 4 and 5, the news and microblog retrieval pipeline of the framework is discussed. Finally, in chapter 6, we summarize the thesis and examine possible future directions.

Chapter 2

RELATED WORK

In this thesis, we propose a unified framework to support event related information seeking. It consists of a news retrieval pipeline and a microblog retrieval pipeline. The literature related to both the retrieval on the genres and the specific techniques used in the pipelines are discussed.

2.1 General Ad-Hoc Retrieval Techniques

Both pipelines perform basic retrieval tasks by leveraging classic ad-hoc retrieval models, such as BM25 [43, 42], language modeling [60], and F2EXP [16]. BM25 is a classic retrieval function which assumes that documents and queries are “bag-of-words”, meaning only the number of occurrences are considered but word orders are not. The relevance probability of a document given a query is decided by the importance of the terms in the query and the eliteness or aboutness of the document to the terms. The aboutness is measured as the term frequency in the document whereas the importance is measured as inverse document frequency described in Equation 2.1. The scoring function of BM25 is shown below:

$$\mathrm{IDF}\left(q_i\right) = \ln\left(\frac{N - n\left(q_i\right) + 0.5}{n\left(q_i\right) + 0.5} + 1\right). \tag{2.1}$$

In the equation, q_i represents one of the n terms in the query and $n(q_i)$ represents the number of documents in the collection containing q_i. N, on the other hand, denotes the number of documents in the collection. Moreover, the score of the document is also normalized by the length of the document with respect to the average length of the documents in the collections, which is considered the standard length of a document. The scoring function of BM25 is shown below:

$$Score_{BM25}(D,Q) = \sum_{i=1}^{n} \mathrm{IDF}\left(q_i\right) \cdot \frac{f\left(q_i, D\right) \cdot \left(k_1 + 1\right)}{f\left(q_i, D\right) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\mathrm{avdl}}\right)}, \tag{2.2}$$

in which Q and D denote the query and the document, respectively. $\mathrm{IDF}\left(q_i\right)$ is computed following Equation 2.1. The term frequency of a term q_i in the document is noted by $f\left(q_i, D\right)$. $|D|$ denotes the length of the document and *avdl* denotes the average document length. k_1 and b are free parameters.

Similar to BM25, language modeling also makes the “bag-of-words” assumption. However, it assumes that a document is generated by a multinomial distribution of words (i.e. language model). The model is estimated by maximum likelihood estimation combined with various smoothing methods. We adopted the smoothing method with Dirichlet prior since existing work seems to suggest it is more effective than other smoothing methods [60]. More specifically, the smoothed model is computed via maximum a posterior estimation by assuming a Dirichlet distribution of the words as the prior, and the Dirichlet distribution is generated by using the term collection frequencies, meaning the appearances of terms in the collection (i.e. $p(q_i \mid \mathcal{C})$). The relevance of a document for a query is decided by the likelihood of the query being generated by the smoothed language model of the document. Following the notations above, the scoring function of language modeling with Dirichlet smoothing is shown below:

$$Score_{LM}(\mathrm{Q}, \mathrm{D}) = \sum_{i=1}^{n} f(q_i, Q) \log \frac{f\left(q_i, D\right) + \mu p(q_i \mid \mathcal{C})}{|D| + \mu}, \tag{2.3}$$

in which $f(q_i, Q)$ is the query frequency of q_i and μ is the free parameter of the Dirichlet prior.

F2EXP is proposed by axiomatic analysis of the existing retrieval models [16]. More specifically, existing models are decomposed into three components, which are Primitive weighting function, Query growth function, and Document growth function.

A set of axioms, or constraints and desirable properties, of the functions, are proposed. More desirable implementations of these functions in terms of the compliance of the axioms are combined to form new retrieval models. F2EXP is one of the models proposed in this way and achieves state-of-the-art effectiveness on multiple benchmark data sets. Its detailed implementation is shown in Equation 2.4 with a similar notation of the previous models. In addition, dfq_i is the document frequency of q_i, whereas s and k are free parameters.

$$Score_{F2EXP}(Q,D) = \sum_{i=1}^{n} f(q_i,Q) \times \frac{f(q_i,Q)}{f(q_i,Q) + s + s \cdot \frac{|D|}{avdl}} \times \left(\frac{N+1}{n(q_i)}\right)^k \tag{2.4}$$

The above mentioned retrieval models are often considered state-of-the-art and are widely adopted by popular Information Retrieval tools for both academic [57, 35, 51] and industry [28] use. Therefore, these methods are investigated as the underlying retrieval functions for both pipelines.

2.2 News Retrieval

2.2.1 News Retrieval for Event Queries

There is little previous work investigating methods dedicated to general news article retrieval. Rather, news article collections are often used to investigate general ad-hoc retrieval techniques [3, 58]. However, there are a few studies regarding news retrieval for event queries for various search tasks. For instance, Bechberger et al. [4] propose a learning-to-rank based method for personalized event news recommendation. Another study that may be more related to background news article retrieval focuses on leveraging the article from one news source (e.g. Reuters) with editorially assigned labels of an event to cluster articles from other news agencies [9]. Although both their work and this thesis attempt to recommend additional news articles given some seed news articles related to an event, one difference is that our method does not involve the help of manual labels of the seed articles. More importantly, the tasks are fundamentally different between these two lines of work. In their work, the objective is to cluster *similar* news for the same event. In this thesis, however, the objective is to recommend news for background, which, according to the definition, might be about *different* but related events.

To the best of our knowledge, this work is the first to specifically design methods for background news retrieval for event related queries. Background retrieval was first formally investigated in the background linking task of the 2018 TREC News Track [47], and subsequently in its iterations in 2019 and 2020. In this task, a system is required to return background articles for a given query article. However, past participants tackled it as a regular ad-hoc retrieval problem. For instance, Ding et al. use different part of the query articles as query and investigate different retrieval methods, such as pseudo relevance feedback with Rocchio or RM3 [1] as well as BERT [14] based result re-ranking [15]. Similarly, Missaoui and MacFarlane try to also use pseudo relevance feedback methods but focus on selecting named entities instead of regular terms [32]. However, our work designs a method dedicated to background retrieval that focuses on using aspects.

2.2.2 Graph Based Method

In previous work, in order to rank text units of documents, such as words or sentences, graphs are built for documents to leverage the associations between them, such as co-occurrence or similarity. For instance, Litvak and Last [27] uses stemmed words as nodes and add direct edges between two words if one precedes another in a sentence. In order to rank the nodes, a supervised method is proposed, which uses features such as in and out degrees of the nodes as well as the frequencies of the corresponding words. Besides, they apply the HITS algorithm [23], which is unsupervised and leverages the incoming and outgoing edge information of the nodes. Mihalcea and Tarau [31], on the other hand, design the TextRank graph-based ranking algorithm, which is similar to PageRank [37] with edge weights taken into account. It is tested on building graphs to rank keywords or sentences in a document. For keyword ranking, a word graph similar to the previous method is built for a document, but undirected edges representing word co-occurrences are used. Edge weights are set to be the same for all edges. Unlike the previous two studies, Boudin [8], instead of proposing new methods, evaluates different centrality measures for ranking keywords with a word graph. TextRank as well as other classic centrality measures such as degree and betweenness are investigated.

The success of the previous work of leveraging word co-occurrences in the form of graph motivates us to use it for aspect identification method as well as aspect importance estimation. Similar to ranking keywords, based on our observation word co-occurrences can also be useful for aspect identification since words belonging to the same aspect are likely to frequently co-occur. However, instead of using any words, in this work, only entities are used since they may be more discriminative and are unlikely to be shared among aspects. Besides aspect identification, the entity graph is also used to weight the entity nodes with centrality measures, which is used for aspect importance estimation.

Specifically for aspect identification, the Louvain method proposed by Blondel et al. [6] is used to segment the entity graph into aspects. This method groups nodes of the graph into communities by maximizing modularity, which is a measure that evaluates partitions of a graph. It rewards edges within the same communities while penalizes those across different communities and the weights of the edges decide the magnitude of the rewards or penalties. This measure reflects our intuition that in the entity graph of a news article, entities of the same aspects should co-occur often (i.e. with high-weight edges in between) whereas entities of different aspects should not co-occur or co-occur less often (i.e. with no edges in between, or the edge weights are low). More specifically, this method first starts with singleton communities and iteratively merges communities with one of their neighbor communities that maximizes the gain of modularity. This process results in small communities, and each of them is grouped into one node and the edges between two communities are merged as one edge whose weight is the sum of the individual weights. The first step as well as this merging step is repeated until no modularity gain can be further achieved.

2.3 Microblog Retrieval

2.3.1 Silent Day Detection

Silent day detection is closely related to adaptive filtering [50] since both of them need to determine whether to deliver any information to users. However, the major difference is that adaptive filtering often deals with long-term information needs that have judged relevant samples. As a result, the commonly used methods often can use the judged data to build an interest model based on an information need. However, in microblog retrieval, the information needs usually are of shorter term interests. Most interests, in particular those event-driven queries, would not last very long. Thus, there is often no training data for the queries, making it difficult to directly apply existing adaptive filtering methods.

Another research problem related to silent day detection is query performance prediction, which predicts the performance of a query in the absence of the relevance judgments [44, 12, 38, 2, 11, 62, 61, 46, 59, 10, 54]. Silent day detection can be tackled by predicting the performance of the set of microblog posts published in a day with respect to the query. However, directly using query performance prediction methods on silent day detection might not be suitable. The scenario of no relevant tweets is often not considered by these methods. Moreover, existing predictors often use single-term based retrieval signals. As mentioned earlier, these signals are not as effective on microblogs and therefore may lead to the mislabeling of silent days as non-silent days. The most similar task to silent day detection might be missing content detection [59]. Nevertheless, it deals with document collections instead of microblog collections.

The concept of silent day was first introduced in the 2015 Microblog Track [25]. The main evaluation metrics in the track heavily penalize systems that return results on silent days. Indeed, previous work [52] shows that the performance of a participating system of the track is dominated by its ability to detect silent days. However, silent day detection was usually tackled *indirectly* by posing a threshold for single tweets, which is either time based [34] or score based [64]. This paper directly addresses the problem of silent day detection.

2.3.2 Pseudo Relevance Feedback

Pseudo relevance feedback (PRF) is a technique used to alleviate the query-document vocabulary mismatch problem in order to improve the retrieval effectiveness. It could be particularly useful in micoblog retrieval due to the short nature of microblog posts. Pseudo relevance feedback assumes that the top retrieved results are relevant and selects expansion terms from them to perform another round of retrieval. Various pseudo relevance feedback methods have been proposed. For instance, Fang et. al. [17] used mutual information between query terms and terms in the top results to select terms that are semantically related to the query. Abdul-Jaleel et al., likewise, propose RM3 which uses the query and top documents in the result to estimate a relevance language model of the query [1]. More specifically, the relevance model $P(w \mid \mathcal{R})$ is estimated based on the query likelihood of the top documents and the documents' language models, which is illustrated below:

$$P(w \mid \mathcal{R}) = \sum_{D \in \mathcal{R}} P(w \mid D). \tag{2.5}$$

The top terms in this model are then added to the query. This method is also selected as a point of comparison since past research has illustrated that it can frequently help to improve the retrieval performances significantly. Although existing PRF methods differ in the way that the scores of the expansion terms are evaluated, their term scoring functions are based on classic retrieval signals. such as term frequency and inverse document frequency. However, in this work, we employed a query collectivity based scoring function to conservatively select expansion terms.

Despite the success that pseudo relevance feedback methods enjoy on document collections, they seem to fail on microblog collections because of the often noisy initial retrieval results [65, 65]. Previous research tackles this problem in various ways. Miyanishi et al. [33] proposed a method that involves manually picking a relevant tweet from top results to select expansion terms from. However, other methods focus on selecting terms from external resources rather than from the target collection for retrieval. A few examples of the external resources are the search results of Google and knowledge bases such as Wikipedia and DBpedia [63, 65, 7]. The rationale of using them is that the information quality is generally higher on these resources than that of a microblog collection so that it is more likely to find relevant information using a basic retrieval function. Nonetheless, in this work, we investigate how to effectively select expansion terms under the settings of PRF where no manual effort is involved and no extra resources are used.

2.4 Bridging News Retrieval with Microblog Retrieval

There have been various attempts to leverage one of the information sources between news and microblog to assist the retrieval on the other. Phelan et al. [40] extract tweets from either Twitter's public timeline or user's Twitter timeline and build an index from the tweets. Another index is built on the articles from user provided RSS feeds. The overlapping terms are then used to retrieve and rank articles from the RSS index. Moreover, Jonnalagedda and Gauch [22] use the popularity of news topics on Twitter to recommend news stories to users. More specifically, tweets are sampled from Twitter and searched on the news collections which assigns cosine similarity scores to the news articles with respect to the tweets. These scores are aggregated for each article to produce the popularity scores of articles, which, in turn, are used to rank the articles. Conversely, a study that is more related to our work tries to link a news article to social media utterances that implicitly reference it, and Twitter is chosen as one of the social media [55]. Multiple queries are generated from the article using its structural elements, such as title, as well as other derived representations, such as named entities. The queries are used to retrieve social media utterances via query likelihood model and constraints are implemented to ensure that the utterances are in close time proximity to the article, and that they are informative. However, unlike this work, we focus on discovering microblogs posts related to the background of news articles, which can be connections to other news stories, instead of referencing them.

Chapter 3

FRAMEWORK DESCRIPTION

In this chapter, we discuss the proposed unified framework for information seeking of event related search interests on news and microblog data. We first offer a high level view of the system, which is shown in Figure 3.1. The components of it which we focus on, and propose novel techniques for, are subsequently discussed in detail.

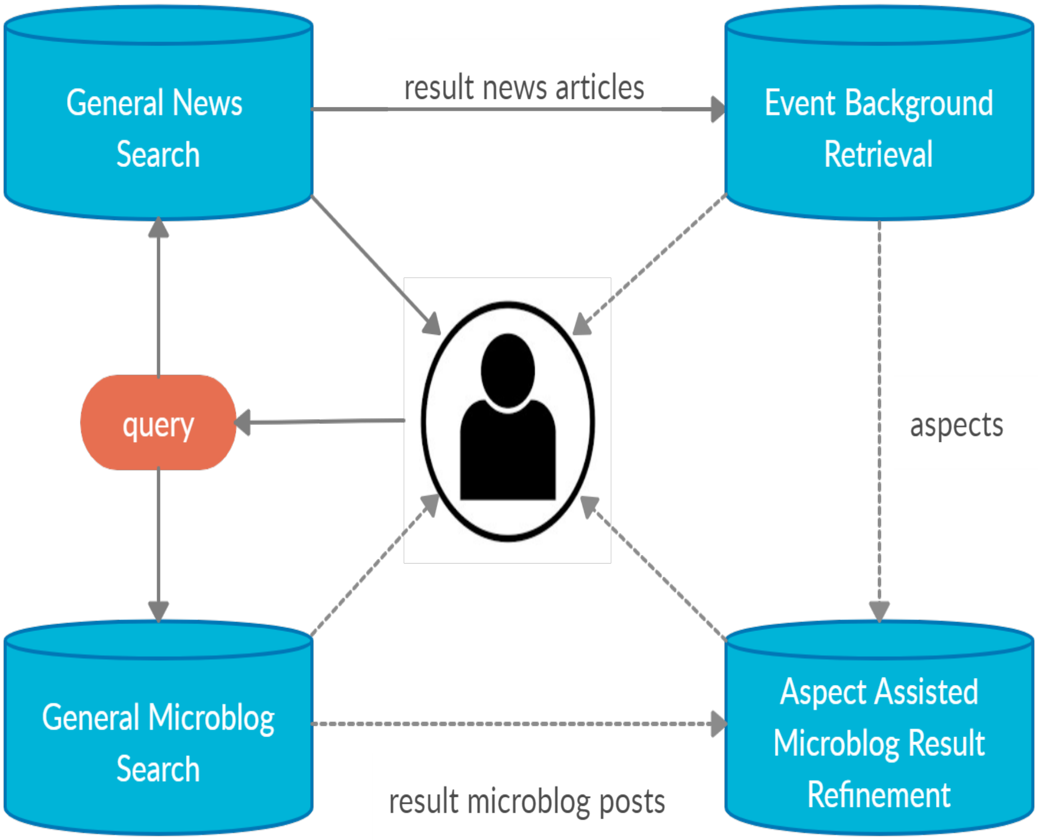
3.1 Framework Overview

As discussed earlier, the framework consists of two parts. The **news retrieval pipeline** is depicted in the top half of Figure 3.1, whereas the bottom half of the figure illustrates the **microblog retrieval pipeline**. It is important to note that the framework is designed in a query type agnostic fashion, which means that it can handle all types of queries regardless of whether they are event related or not. This also results in the fact that not all paths of the system shown in the figure are taken for all queries. The paths shown as dashed lines are only taken by some queries under certain circumstances, which is discussed below.

In the news retrieval pipeline, the “General News Search” step is performed by classic ad-hoc retrieval techniques, and the results are sent back to the users. In addition, the result articles are sent to the “Event Background Retrieval” module in which event articles are identified. If there are event articles among the result articles, suggesting the query is event related, background articles are retrieved are returned to the users.

On the other hand, in the microblog retrieval pipeline, silent day detection and pseudo relevance feedback can be applied in the first step “General Microblog Search”. Results are returned to the users if the search is ad-hoc or, for daily searches, if the days

Figure 3.1: Overview



are non-silent. If the query is event related according to the news retrieval pipeline, the retrieved microblog data are re-ranked and re-organized for different aspects of the events and shown to the users.

3.2 News Retrieval Pipeline

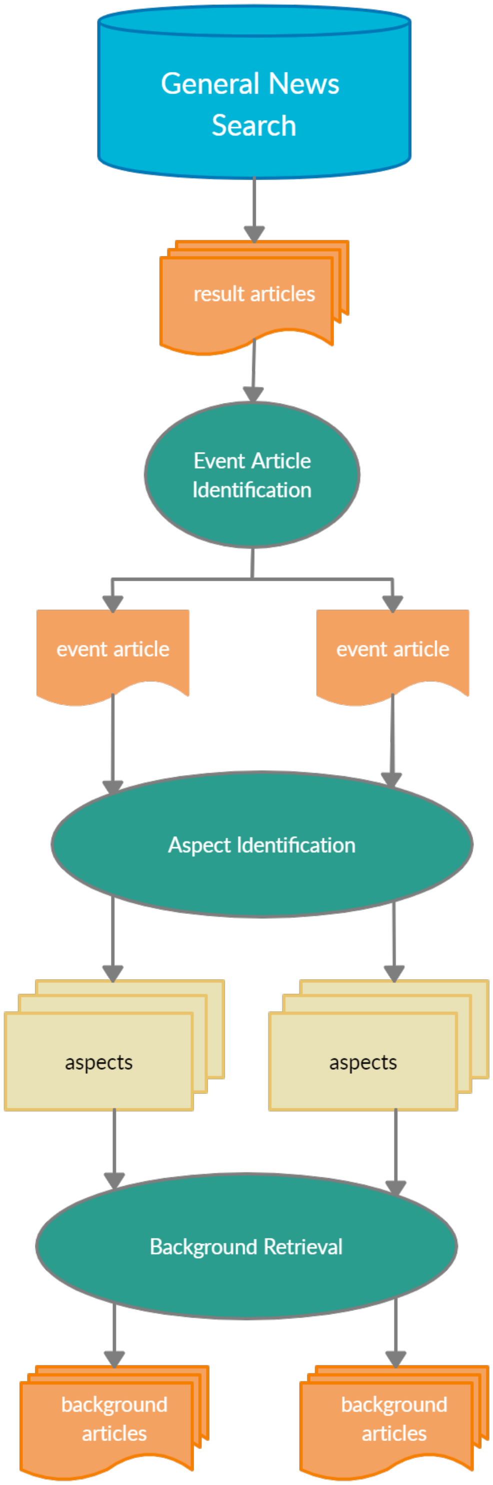
In the news retrieval pipeline, we focus on "Event Background Retrieval" and the workflow in it is shown in Figure 3.2. The event articles are identified by the event article identification method from the results of the first retrieval step. Subsequently, if there are any event articles, they are used for background retrieval and non-event articles are ignored. This is because, as mentioned earlier, we decide to use the mentioning of background information in event articles as a guide to retrieve further details of the mentioned background information. Since there can be multiple aspects of background, the aspects are identified in the event articles. These aspects are individually used to find articles of background information. Results from different aspects of the same event article are merged. However, it is important to note that the background articles for different event articles are not. The primary reason for it is that there is no appropriate testing data. However, as shown experimentally in the microblog retrieval pipeline (which will be discussed later), we propose a reasonable way to weight aspects across different event articles so that the weights are comparable. Besides, retrieving background articles on a per event article basis may have its own benefits too. As argued before, the need of extra background information may be developed when a user reads background aspects in an event article, and he or she wants to know more about them. As a result, displaying the links to the background articles about these aspects in the sidebar of the event article would be more convenient for the user than aggregating background articles from multiple event articles. This is because the latter method might include background articles not related to the aspects in the event article the user is reading, which are not of *immediate* interests of the readers. In fact, displaying background articles in the sidebar of an event article is the use case of the background linking task of the News Track [1], whose data the proposed background retrieval framework is tested on.

3.3 Microblog Retrieval Pipeline

Besides the news retrieval pipeline, the other half of the proposed framework is the microblog retrieval pipeline. The first component of it is called "General Microblog Search", and the detailed workflow is shown in Figure 3.3. Depending on the temporal characteristic of the search, microblog data of different time ranges are searched on. If this search is ad-hoc, it is performed on things published moderately recently (i.e.

[1] http://trec-news.org/

Figure 3.2: Event Background Retrieval



within several days in the past) on the microblog and the retrieved microblog data is returned and the subsequent steps proceed if needed. However, if the search is performed daily, the system checks whether the day is silent before returning any results to the users. If it is, no results are returned. Otherwise, the search is performed on the microblog published within the current day, and the following steps are identical to that of the ad-hoc microblog search. In both search scenarios, the query collectivity based measures can be used to perform pseudo relevance feedback to further improve the search effectiveness. It is important to note that the first component of the microblog retrieval pipeline can be useful for not only event related but also non-event related queries since the methods we proposed for the component are designed for general microblog search.

However, the second component of the pipeline "Aspect Assisted Microblog Result Refinement" is designed specifically for event related queries and the workflow is shown in Figure 3.4. If the system decides to return microblog results to a user for a query and there are event articles in the news results, the aspects extracted from the event articles are used to re-rank and re-organize the microblog respects on a per-aspect basis to offer users the option of picking the aspects they want to explore.

Figure 3.3: General Microblog Search

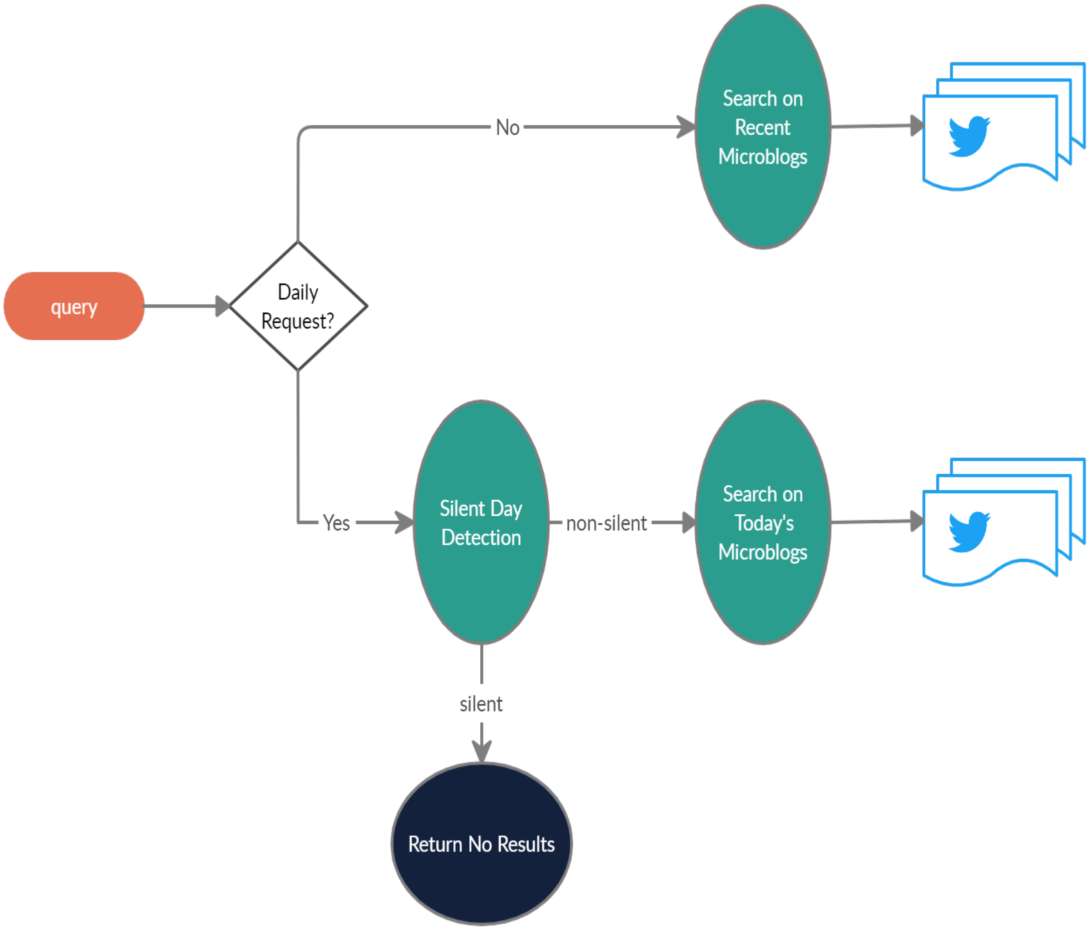
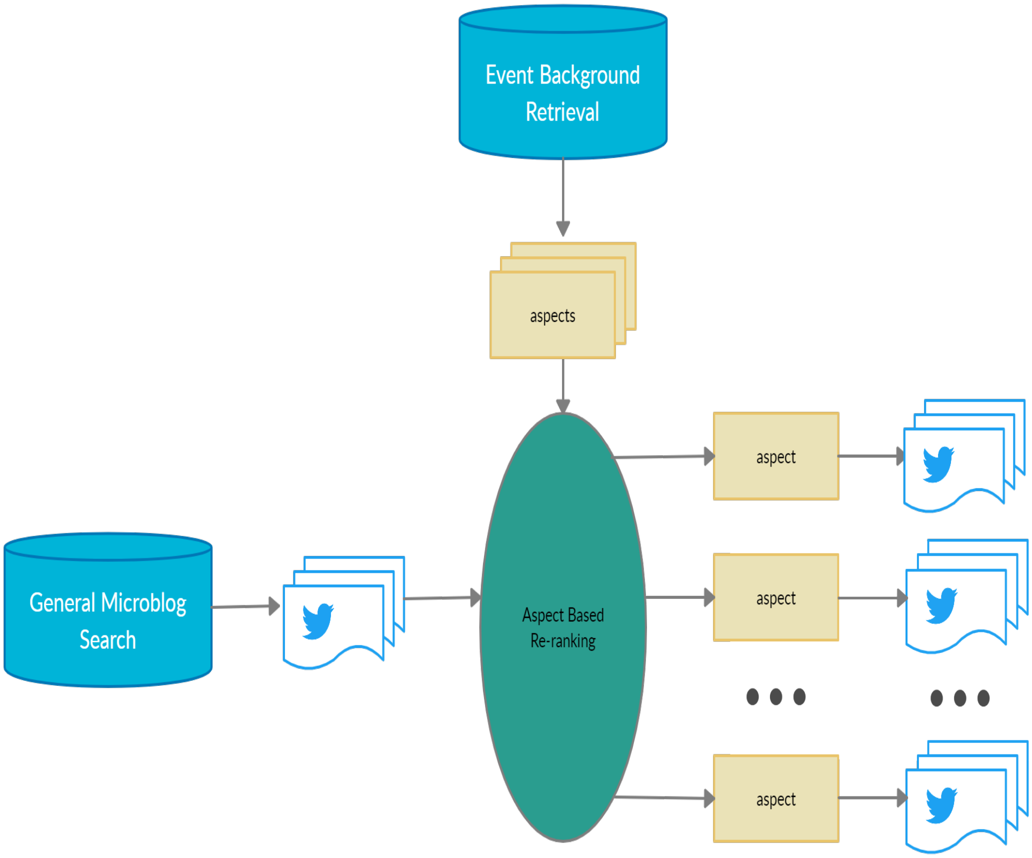


Figure 3.4: Aspect Assisted Microblog Result Refinement



Chapter 4

BACKGROUND INFORMATION RETRIEVAL FOR EVENT RELATED QUERIES

In this chapter, we discuss the background retrieval part of the news retrieval pipeline which consists of event article identification and background retrieval. For event article identification, a simple yet effective rule based method is designed which uses article titles. If event articles are detected in the retrieved articles, background retrieval is applied. In order to effectively retrieve background articles, based on the temporal and multi-faceted nature of the background information, two sets of methods are proposed. From the temporal perspective, a simple yet effective time filter is applied to constrain the retrieved background articles to be published before the corresponding event articles. On the other hand, according to the multi-faceted nature of the background information, we propose a probabilistic model that retrieves background articles depending on the aspects they cover and the importance of the aspects. Since news retrieval has been intensively studied in the literature, we leverage existing techniques for retrieving news articles for aspects while focusing on how to identify background aspects from event articles, and how to weight the aspects. An entity graph based method is proposed to identify aspects whereas different ways of weighting aspects are designed based on how clearly they are explained in the event articles or their relatedness to the event being reported. The proposed methods are tested on the News Track data sets and analysis is conducted to examine the overall and each component's effectiveness.

4.1 Methodology

4.1.1 Identifying Event Articles

Based on our observation, a rule based method is designed to use the title of the articles. The rules are described as follow:

- An article is classified as an event article if there is at least one capitalized word in the title, and the word is consistently capitalized in the body text.
- An article is classified as an event article if the verbs in the title are in the past tense.

The title is used since it is often a short summary of the main content of the article. As a result, in an event article, the title often briefly describes the event. Our rule based method reflects our observation of how the event is described in the title. The first capitalized word rule is motivated from our observation that, in an event article, *who* and *where* of the “five w's” of the news often contain capitalized words and are often mentioned in the titles. Since some words are capitalized for grammatical reasons, such as the first word of the title, a capitalized word can only be considered if it is consistently capitalized in the body of the article. In addition to the capitalized word rule, the verb tense is also considered because an event sometimes is reported in the past tense. This rule-based method is simple and therefore can be efficiently executed. Its effectiveness is tested in the experiment section.

4.1.2 A Probabilistic Model for Aspect Based Background Information Retrieval

As mentioned earlier, given an event, background information is multi-faceted. In other words, there can be multiple *aspects* of an event and the background information of the event can be about any one of them. Based on the definition of background information, mentioned previously, we define an aspect of background information of an event as either an aspect of the details of the event, or another story related to the event being reported. Two sets of examples of background aspects and their corresponding background articles are illustrated in Figure 4.1. The event article of the first set of examples in Figure 4.1a is about new species are created as a result of the mating between grizzlies and polar bears [1], and one of its background aspects and the corresponding background article is about the interbred between Humans and Neanderthals [2]. In the second set of examples, the event article is about an incident where a truck driver was indicted for a human smuggling case at the Mexican border after many undocumented immigrants died in the trailer of his truck [3]. An aspect of the background is the details of the suffering of the immigrants in the trailer, which corresponds to the background article about one of the immigrants [4]. Another aspect is the increased difficulty of illegal border crossings under the Trump administration of the U.S., which corresponds to a background article that describes Trump's immigration policies [5]. As can be seen, the first pair in the second example set is an example of the aspect being the details of the event, whereas the other example pairs are other news stories related to the event being reported.

To account for the multi-faceted nature of the background information, a probabilistic model is introduced, which is inspired by a result diversification framework xQuAD [45]. It is motivated by the intuition that, for an article D to contain background information of an event article Q with respect to an aspect a_i, not only does D need to discuss the aspect a_i, but also D has to be related to Q. More specifically, for each aspect a_i and a document D, the probability $Pr(D|a_i)$ can be computed, which measures the relevance of D given a_i using the classic query likelihood estimation. We call it document aspect likelihood. To ensure that D is related to Q to ensure that D discusses the aspect in a similar sense as it is in the event article, document relevance $Pr(D|Q)$ is combined with the document aspect likelihood for ranking documents for

[1] https://wapo.st/30InE9K

[2] https://wapo.st/3fVXZ3J

[3] https://wapo.st/3hwr6uW

[4] https://wapo.st/3huWycP

[5] https://wapo.st/2OQTueP

a specific aspect. Similar to xQuaD, the linear interpolation of the likelihoods is used:

$$\beta Pr\,(D|Q) + (1-\beta)Pr(D|a_i), \tag{4.1}$$

where β regulates the influence between the event article and its aspects.

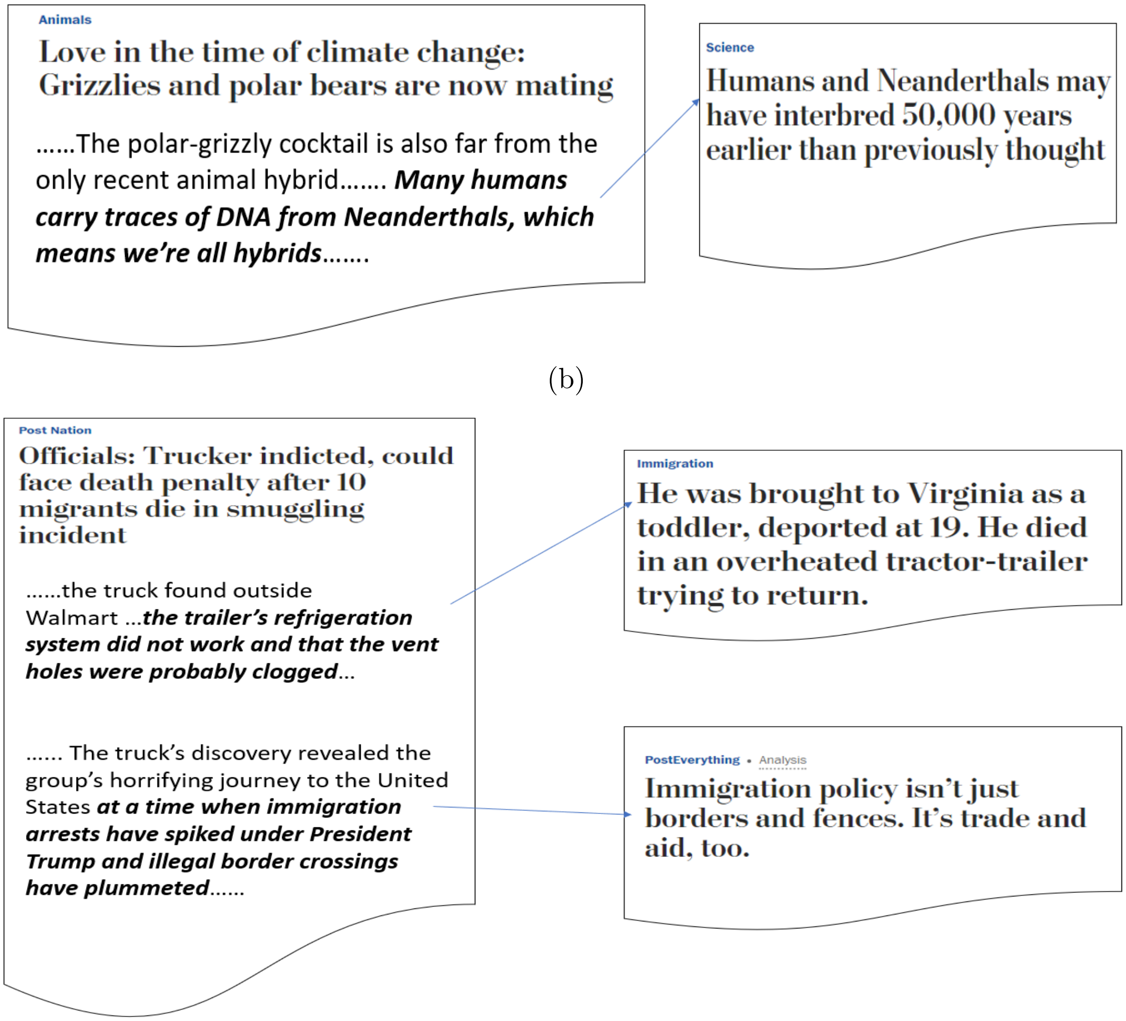
However, the background articles for aspects need to be merged to form a single ranked list of background documents for the event article. Thus, we introduce $Pr(a_i|Q)$ to indicate the importance of the aspect a_i. With this importance component and the aspect document ranking component mentioned above, we propose the following probabilistic model for ranking background documents:

$$Pr(B=1|D,Q) \overset{\text{def}}{=} \sum_{a_i \in Q} Pr(a_i|Q)(\beta Pr\,(D|Q) + (1-\beta)Pr(D|a_i)). \tag{4.2}$$

In the above equation, B is a binary random variable indicating whether a document D is a background article (1) or not (0). This model simulates the process where a user reads an event article, discovers the aspects along with the importance of them presented by $P(a_i|Q)$, and rates/ranks other documents based on the aspect importance as well as how likely the document contains information about the aspects.

The above method is inspired by xQuaD, a result diversification framework, since result diversification and background information retrieval are similar in that they attempt to retrieve information for different aspects. Nevertheless, there are major differences between them in terms of the problem setup. The most salient distinction is that, in result diversification, the decision of whether a document needs to be retrieved depends on the documents that are already retrieved and a system prioritizes documents covering new aspects that are not clearly discussed in the previously returned documents. In background information retrieval, however, such dependence is not assumed. The ranking of the documents only depends on how well the documents covering the background aspects and how important are these aspects. Admittedly, it would be desirable to diversify background information. Moreover, the proposed

Figure 4.1: Examples of background articles



(a)

Animals

Love in the time of climate change: Grizzlies and polar bears are now mating

……The polar-grizzly cocktail is also far from the only recent animal hybrid……. ***Many humans carry traces of DNA from Neanderthals, which means we're all hybrids***…….

Science

Humans and Neanderthals may have interbred 50,000 years earlier than previously thought

(b)

Post Nation

Officials: Trucker indicted, could face death penalty after 10 migrants die in smuggling incident

……the truck found outside Walmart …***the trailer's refrigeration system did not work and that the vent holes were probably clogged***…

…… The truck's discovery revealed the group's horrifying journey to the United States ***at a time when immigration arrests have spiked under President Trump and illegal border crossings have plummeted***……

Immigration

He was brought to Virginia as a toddler, deported at 19. He died in an overheated tractor-trailer trying to return.

PostEverything • Analysis

Immigration policy isn't just borders and fences. It's trade and aid, too.

aspect based framework arguably fits nicely for the purpose. However, due to the lack of data for evaluation under such a setup, we leave if for future work.

After defining the model, how different components are implemented needs to be decided. Two components document relevance $Pr(D|Q)$ and document aspect likelihood $Pr(D|a_i)$ can be implemented using existing retrieval techniques such as language modeling [49, 60]. Thus, our focus is on other parts of the model, which are the aspect identification and aspect importance estimation.

4.1.3 Entity Graph Based Aspect Identification and Aspect Language Model Estimation

It is clear that the aspects are a crucial part of our framework. Only when the aspects are accurately identified can the framework be effective. Since we approach retrieving background information for aspects as a classic retrieval task, “queries” of the aspects, meaning some bag-of-words representations, are required. Therefore, we specifically try to estimate the language models for the aspects. It can be easily plugged into classic retrieval frameworks such as language modeling, and it has weights associating with different words, which could be helpful for the retrieval step.

In order to identify the background aspects in the event articles, how background information is written needs to be analyzed. Sometimes, a reporter writes aspects in one or more paragraphs separated from the reporting of the event. However, more frequently, background information is blended with the main story by a technique called information weaving [19]. By doing so, background information is introduced alongside the reporting of the mains story without breaking the flow of the story. The second aspect in Figure 4.1b is an example of that as the background is explained in a subordinate clause while the main clause is about the main story. Therefore, we introduce an entity graph based method to tackle the task. It is important to note that we define the term “entities” here as an umbrella term that encompasses both named entities (such as Trump and Walmart in the above truck driver indictment query article) and concepts (such as “Humans” and “Neanderthals” in the bear interbred query article) that have Wikipedia pages associated with them. This is because using only named entities might be insufficient and missing important concepts in identifying aspects, such as “Humans”. It is important to note that we use the toolkit called DBpedia Spotlight [13] for entity annotation and not *all* words or phrases with associated Wikipedia pages are annotated due to the filtering steps of the toolkit. For a phrase or a word in the input text (which is called surface form), a list of candidate entities in Wikipedia is generated. The tool then computes the similarity scores between the text around the surface form in the input text and the text surrounding the candidate entities in

Wikipedia. No annotations are generated if all candidate entities' similarity scores are low. Other criteria for filtering are considered as well, such as how common the entities are and whether the difference of the scores of the first and second ranked candidates is big enough (i.e. whether the best annotation is ambiguous). The filtering steps are desirable since they can improve the entity annotation quality.

Our observation of the entities in an article with respect to aspects is that different aspects tend to use different sets of entities and that there is little overlapping between the entities of different aspects. Nouns, verbs, and other words that are not as discriminative as entities, however, are more likely to be shared among aspects. For instance, regarding the truck driver indictment event article, the entity "Trump" is only used by the aspect describing immigration policies in the Trump era, whereas the entities such as "Walmart" and the drug cartel "Los Zetas" are used to describe how the immigrants were smuggled into America as well as the terrible conditions they suffered. On the other hand, non-entity words such as "immigrants" might be shared by aspects. Thus, we could separate entities in the event article into several non-overlapping groups, and each group represents an aspect of the article. Nevertheless, it is not reasonable to use all entities in the query to identify aspects because, besides aspects, entities are used in describing the main event of the article as well. These entities need to be removed before the segmentation of the entities into aspects. Intuitively, the entities of the event may occur frequently in the article throughout the whole document. Entities only appear frequently in a small region of the article might be those belonging to aspects. Thus, we use paragraph frequency, which is the number of paragraphs in which the entity appears, as the measure to pick the entities for the event. A natural way of using the paragraph frequency is to rank entities with paragraph frequencies, and remove top entities that are likely used for describing the event. However, the usage of entities varies among event articles based on the events they are reporting. Some events require the use of more entities. Other events may use nouns more often, such as "immigrants" and "truck" in the trucker indictment case. Therefore, we rank nouns and entities together by paragraph frequency and heuristically only remove the entities among the top five of the ranked list for identifying aspects.

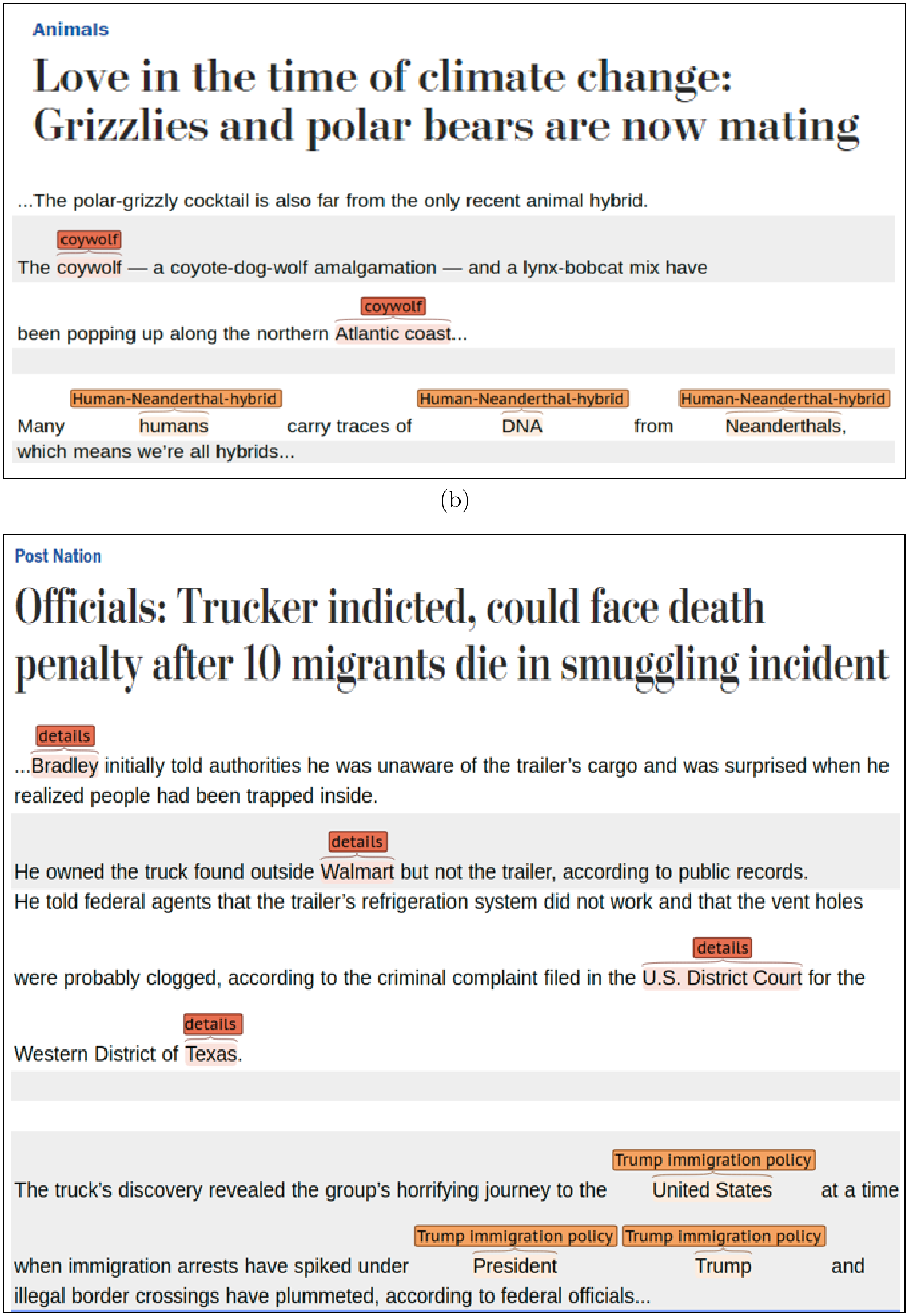
With entities about the event being removed, the next step is to segment the remaining entities into different aspects. Intuitively, entities of the same aspect often co-occur in the same paragraphs, and the opposite is often true for entities from different aspects. Thus, to leverage the entity co-occurrence information, we follow previous work [27, 31] and build an entity graph from aspect entities of each event article. In the graph, the nodes are the entities while the edges represent the co-occurrence relations of the entities in paragraphs. Moreover, the edges are assigned weights according to the word distances between the entities in different paragraphs. More specifically the weight between two entities e_1 and e_2 in a paragraph p that they co-occur in is computed as:

$$W(e_1, e_2, p) = 1 - \frac{1 + \#\text{ of words between } e_1, e_2}{|p|}, \tag{4.3}$$

where $|p|$ is the word count in the paragraph. It is used so that the weight is normalized and comparable between paragraphs. The edge weight between two entities is the average of the above weights among all paragraphs that they co-occur. The Louvain method proposed by Blondel et al. [6] is used to finally separate the entities into aspects. This method is designed for community analysis, which attempts to separate a weighted network into densely connected communities/sub-graphs. It accomplishes this by approximately optimizing modularity, which measures how edges are concentrated within communities instead of between them. It is clear that the method can segment entities following our intuition of grouping the entities co-occur often into the same aspects. In Figure 4.2, two example articles and their aspects are shown. More specifically, the aspects are illustrated in the form of highlighted entities. Entities belonging to different aspects are marked with different colors, and we provide a short description for each aspect for illustration purposes.

After aspects are identified, the language models of them can be estimated. Using the mined aspects from the entity graph, given a paragraph, it can be labeled with the aspects whose entities occur in the paragraph. The language model of an

Figure 4.2: Examples of aspect entities in articles



(a)

Animals

Love in the time of climate change: Grizzlies and polar bears are now mating

...The polar-grizzly cocktail is also far from the only recent animal hybrid.

The coywolf — a coyote-dog-wolf amalgamation — and a lynx-bobcat mix have been popping up along the northern Atlantic coast...

Many humans carry traces of DNA from Neanderthals, which means we're all hybrids...

(b)

Post Nation

Officials: Trucker indicted, could face death penalty after 10 migrants die in smuggling incident

...Bradley initially told authorities he was unaware of the trailer's cargo and was surprised when he realized people had been trapped inside.

He owned the truck found outside Walmart but not the trailer, according to public records. He told federal agents that the trailer's refrigeration system did not work and that the vent holes were probably clogged, according to the criminal complaint filed in the U.S. District Court for the Western District of Texas.

The truck's discovery revealed the group's horrifying journey to the United States at a time when immigration arrests have spiked under President Trump and illegal border crossings have plummeted, according to federal officials...

aspect then can be estimated from the paragraphs belonging to the aspect. There might be two ways of achieving this. One of them is called Labeled Latent Dirichlet Allocation (L-LDA) [41]. It is similar to the classic LDA method. The only difference is that, instead of assuming words in a document can be generated from all the topics, a document is labeled by a subset of topics and it can only draw topics from the set when generating words. Under the setting of aspect language model estimation, paragraphs correspond to "documents", and aspects correspond to "topics" in L-LDA, respectively. Following the procedures of L-LDA, the language models for the aspects can be estimated. Besides aspects, the reporting of the event should be considered as well. Due to the commonality of the usage of the information weaving technique, we also assume that the reporting of the event co-exists with all the aspects and assign the label of the event to all paragraphs.

In addition to L-LDA, we propose another way of leveraging aspect labels of the paragraphs to estimate aspect language models that is based on Probabilistic Latent Semantic Analysis (PLSA) [21, 29]. The method is inspired by L-LDA, and it is assumed that the words of the paragraphs are generated by a mixture model of the aspect models, the event model, and the collection model [6]. However, the aspect models are constrained to be only those that correspond to the aspects of the paragraphs. As a result, the method is called *Labeled PLSA* (L-PLSA). We introduce the notations for the proposed method. More specifically, we denote a_i as the i-th aspect, and, for the sake of cleaner presentation, we denote the event as a special aspect a_0. Given a paragraph p, the count of word w in it is noted as $c(w, p)$, and its aspects labels are noted as L_p. Following the settings of the previous method, we assume that each paragraph can be generated from the event model, which means $0 \in L_p \ \forall p$. θ_i is the language model of aspect a_i and θ_C is the collection model. A hidden variable $z_{p,w}$ denotes the aspect model that a word w in paragraph p is generated from. $Pr(z_{p,w} = i)$ is the probability that the word is generated by θ_i, while $Pr(z_{p,w} = C)$ is the probability that the word

[6] In the previous work [29], this is called "background model" and is estimated from the whole collection. We call it the collection model to avoid confusion.

is generated from the collection model. $\pi_{p,i}$ is the mixing weight for paragraph p to choose θ_i for word generation and $\sum_{i \in L_p} \pi_{p,i} = 1 \ \forall p$. λ_C is the mixing weight for θ_C. Following the previous work on PLSA [29], we use the EM algorithm to estimate the aspect language models θ_i as follows:

E-Step:

$$Pr\left(z_{p,w} = i\right) = \begin{cases} \frac{\pi_{p,i}^{(n)} Pr^{(n)}(w|\theta_i)}{\sum_i \pi_{d,i'}^{(n)} Pr^{(n)}(w|\theta_{i'})} & i \in L_p \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$Pr\left(z_{p,w} = C\right) = \frac{\lambda_C Pr\left(w \mid \theta_C\right)}{\lambda_C Pr\left(w \mid \theta_C\right) + \left(1 - \lambda_C\right) \sum_{i \in L_p} \pi_{p,j}^{(n)} Pr^{(n)}\left(w \mid \theta_i\right)} \tag{4.5}$$

M-Step:

$$\pi_{p,i}^{(n+1)} = \begin{cases} \frac{\sum_w c(w,p)(1-Pr(z_{p,w}=C))Pr(z_{p,w}=i)}{\sum_{i' \in L_p} \sum_w c(w,p)(1-Pr(z_{p,w}=C))Pr(z_{p,w}=i')} & i \in L_p \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$$Pr^{(n+1)}\left(w \mid \theta_i\right) = \frac{\sum_p c(w,p)\left(1 - Pr\left(z_{p,w} = C\right)\right) Pr\left(z_{p,w} = i\right)}{\sum_{w'} \sum_p c\left(w', p\right)\left(1 - Pr\left(z_{p,w'} = C\right)\right) Pr\left(z_{p,w'} = i\right)} \tag{4.7}$$

The above calculation is very similar to that of the original PLSA and the major differences are that, when computing $Pr(z_{p,w} = i)$ and $\pi_{p,i}^{(n+1)}$ for a paragraph p, they are set to zero if p does not have aspect a_i assigned to it (e.g. the entities of a_i do not occur in p). Moreover, when performing normalization for them, they are only marginalized over the set of aspects of p, which is L_p.

4.1.4 Aspect Importance Estimation

Another component of the proposed aspect based background information retrieval model that is the focus of this work is to estimate the aspect importance $Pr(a_i|Q)$. The simplest way is to assume the weights of all aspects are the same:

$$P(a_i|Q) = \frac{1}{|A|}, \tag{4.8}$$

where $|A|$ is the size of all the aspects (i.e. $A = \{a_1, \cdots, a_k\}$). Beyond this, we argue that there can be two interpretations of the aspect importance, and we explain them and their possible realizations. The first interpretation can be called "clarity", meaning how clearly an aspect is discussed or described in the event article. Intuitively, if an aspect is not explained clearly, the readers may certainly want more details and contextual information about the aspect. However, directly measuring clarity might be extremely challenging. It is difficult to mathematically define and measure the clarity of an aspect in the event article. More importantly, the clarity requirements of different aspects might be different. The aspects that are more familiar to the general public, such as Trump's immigration policies, might require less explanation for readers. In other words, the clarity requirements for them are lower. Less known aspects, such as the details of the sufferings of the immigrants, might require a higher level of clarity. Thus, we employ a simple and heuristic method and leave exploring more sophisticated techniques in future work. More specifically, we assume that the familiarity of all aspects to a user is the same prior to him or her reading it, and we use the amount of content in the event article corresponding to the aspects to measure their clarity. The intuition behind using the amount of text is that, the fewer words used to describe an aspect, the more likely readers do not fully understand the aspect, and the more important it is to provide background information for the aspect. This amount can be measured by how many words in the event article are used to describe the aspect via the word-aspect assignment probability $Pr(z_{p,w} = i)$ in L-PLSA. We call this measure the *coverage* of aspects. Given an aspect a_i, its coverage can be computed as:

$$Coverage(a_i|Q) = \sum_{p} \sum_{w \in p} Pr(z_{p,w} = i) c(w, p). \tag{4.9}$$

Since the clarity is negatively correlates to the coverage, the clarity based aspect importance can be computed as:

$$Pr(a_i|Q) = \frac{1 - Coverage(a_i|Q)}{\sum_{a_i'} (1 - Coverage(a_i'|Q))}. \tag{4.10}$$

One is added to the negative of the coverage to ensure the value remains positive, whereas the clarity based importance is normalized over all aspects to ensure $\sum_{a_i \in A} Pr(a_i|Q) = 1$.

Another interpretation of the aspect importance can be called *relatedness*, meaning how closely related an aspect is to the event being reported. Aspects that are weakly related to the event might be insignificant and do not need to be understood fully for the readers. An example of this can be the Trump immigration policy aspect in the truck driver indictment for a human trafficking case. If an aspect is closely related to the event, however, it might be essential to provide background information for the aspect. An example of this could be the details of the sufferings of the immigrants in the above case. A natural way of measuring relatedness could be using co-occurrence. In order to do that, we marginalize the aspect component over documents:

$$P(a_i|Q) = \sum_{D} P(D|Q)P(a_i|Q, D). \tag{4.11}$$

In theory, the marginalization should be done over all documents. However, it would be prohibitively expensive. Moreover, existing work shows that by retrieving top documents using some retrieval function and perform marginalization on this small set of documents can offer reasonably effective performance [17, 1]. Thus, we also adopt this paradigm.

For each document D, the right side of Equation 4.11 represents the likelihood of D being observed given the event article Q, and the likelihood that the aspect a_i is also in D. For the first part $P(D|Q)$, we employ the language modeling approach by performing maximum likelihood estimation on Q to obtain its language model and compute the generative probability of D given the language model. For the second part, we apply Bayes Rule and assume that the aspect prior $P(a_i)$ follows a uniform distribution:

$$\begin{aligned} P\left(a_i \mid Q, D\right) &= \frac{P\left(Q, D \mid a_i\right) P\left(a_i\right)}{\sum_{a_i'} P\left(Q, D \mid a_i'\right) P\left(a_i'\right)} \\ &= \frac{P\left(Q, D \mid a_i\right)}{\sum_{a_i'} P\left(Q, D \mid a_i'\right)}. \end{aligned} \tag{4.12}$$

The core part of it, which is $P(Q, D|a_i)$, can be computed as the probability of given the language model θ_i of aspect a_i, the overlapping words of Q and D are generated, which can be easily computed given the fact that θ_i is estimated previously.

In addition, the entity graph of the event article can be used to compute the relatedness of the aspects to the event as well. In the entity graph of an event article, entities are used to represent aspects and the edges between entities indicate how closely related the entities are. Therefore, we insert the event into the graph in the form of nodes, and use the connections between these nodes and entity nodes of the aspects to measure the relatedness. More specifically, the entities and nouns with the highest paragraph frequencies, which are removed for constructing the entity graph, are put back into the graph. The edges among themselves as well as those to the aspect entities are added back as well. Since the newly added nodes can be either entities or nouns but belong to the event, we call them event nodes. Intuitively, the relatedness between an aspect and the event is high if the aspect's entities are connected to the event nodes, especially the more important ones. We realize the intuition by the equation below:

$$R(a_i) = \sum_{e \in a_i} \sum_{n \in o} \mathbb{1}(e, n) I(n). \tag{4.13}$$

In the above equation, aspect and event are noted as a_i and o, respectively. Their nodes, in turn, are noted as e and n. $\mathbb{1}(e, n)$ is an indicator function whose value is one only when there is an edge between e and n. Otherwise, its value is zero. The importance of n is noted as $I(n)$. $R(a_i)$ denotes the relatedness of the aspect a_i. The node importance can have multiple implementations. The word count in the event article can be used. Besides, the graph structure can be leveraged. Following previous work, centrality based measures, such as degree, closeness, and betweenness, can be used [27, 8]. Some of the measures, such as closeness and betweenness require edge distance instead of edge weights. Since the weights are in the range of 0.0 to 1.0. One minus the edge weight is used as the edge distance. Moreover, TextRank, which is proposed by Tarau and Mihalcea [31] for weighting keywords in a document using the graph structure of a word co-occurrence graph, can also be leveraged. We employ TextRank as well as several centrality based node importance measures, namely betweenness, closeness, degree, and eigenvector. The effectiveness of these implementations is investigated experimentally.

4.1.5 Temporal Perspective of Background Information

In addition to the multi-faceted nature of the background information, the temporal perspective of background information is also analyzed and used to assist the retrieval of background information. When considering background articles of an event article, intuitively they are more likely to be information in the past, and therefore are likely to exist in articles published before the event article. For example, the background articles of a report of the new development of a major criminal case might be articles of the origin as well as the previous development of the case. Thus, we propose a simple time filter to ensure the retrieved results to be articles published before the event article.

4.2 Experiment

The proposed news retrieval pipeline is investigated under the settings of the background linking task in 2018 and 2019 TREC News Track [47], which is designed specifically to investigate how to fulfill the information need of background information. More specifically, we investigate the effectiveness of the rule based event article detection method as well as the two proposed methods that are designed to facilitate the retrieval of background information, which are the time filter and the aspect based background information retrieval model.

4.2.1 Data and Experiment Setup

Both years of the News Track have a common task called background linking. Unlike ad-hoc retrieval settings that use keyword queries, this task provides a list of query articles and requires a system to retrieve articles containing background information given a specific query article. It is clear that such a setup is desirable to test the proposed background retrieval method on, but the query articles are not restricted to event articles. This also enables us to test the event article detection method on the tracks' data. Since there are no appropriate keyword queries associated with the event articles, the proposed rule based method is tested on how accurately it can detect event articles. Moreover, as argued before, if a relevant article of a keyword query is an event article, it is likely that the query article itself is event related.

The two years' of the track have 50 and 60 background linking query articles for 2018 and 2019, respectively. Moreover, they use the same news collection which consists of 608,180 news articles published in the Washington Post from January 2012 through August 2017 [7]. Duplicate documents is a known issue of the collection and we follow the de-duplication procedure proposed by Bimantara et al. [5]. Document files are processed in lexicographic order of their file names, and if an article has the same title, was published on the same date, and shares the same authors as an article encountered previously, this article is considered duplicate and discarded. It is important to note that the track organizers also provide a script for de-duplicating documents, but we follow the above procedure since it provides better effectiveness. After this step, documents published in the sections of "Opinion", "Letters to the Editors", and "The Postś View" are also excluded following the track guideline. These types of articles are not considered background since they are opinion pieces instead of focusing on presenting factual information. Because the proposed aspect based background retrieval framework requires entity annotations of Wikipedia entities, we use DBpedia Spotlight [13] to annotate entity mentions in title and content (which

[7] https://trec.nist.gov/data/wapost/

is document body) fields. Besides detecting the text string of entity mentions, the toolkit also provides the Wikipedia titles correspond to the entities, which are called canonical forms. Since the same entity can be mentioned in different ways in the text, we replace all entity mentions in the original documents with their canonical forms to avoid the mismatching problem. Moreover, the canonical forms can be multi-words, and thus each canonical form is treated as a single word at index time. The processed title and content fields as well as the publication timestamps are indexed using the Indri toolkit [51]. The official metric ndcg@5 is adopted for evaluation.

4.2.2 Event Article Detection

The effectiveness of the event article detection is investigated first since the latter background retrieval step depends on the results of it. We manually label the 110 query articles and there are 66, or 60%, event articles. The numbers of event and non-event articles are reasonably balanced, which is desirable for evaluation. Evaluating the output of the proposed rule based event article detection method against the manual labels, an f1 score of 0.84 is observed on event articles with the precision and recall being 0.81 and 0.86, respectively. This illustrates that the method is not only simple but also effective.

4.2.3 Method Order and Retrieval Baseline

There are two sets of methods proposed for background retrieval for event articles: the time filter and the aspect based background retrieval framework. Since they do not depend on each other, it is important to define the order they are applied. Due to efficiency concerns, the aspect based framework is implemented in a re-ranking manner. More specifically, some methods are first used to perform a round of retrieval to obtain a reasonable pool of background article candidates. These candidates are subsequently re-ranked based on the proposed framework. Thus, it is reasonable to apply the time filter first since it can be efficiently executed.

A retrieval baseline is required to apply the time filter on and there are two essential components of the retrieval baseline which should be decided: the "query" and the retrieval model. We use all the words in the event article, meaning all entities and non-entity words, as the query and investigate the proper retrieval model settings. The all word query is chosen since it is simple yet achieving reasonably effective performances on the background linking task of past News Track's. Several retrieval models are tested, namely f2exp [16], bm25 [43] and language modeling [60]. We perform a grid search on their parameters and evaluate the effectiveness of these parameter settings on all queries. More specifically, s in f2exp is tuned within the range of $[0.1, 1.0]$ with a step size of 0.1; b in bm25 is tuned within the range of $[0.1, 1.0]$ with a step size of 0.1; and μ in language modeling with Dirichlet smoothing is tuned within the range of $[500, 5000]$ with a step size of 500. Language modeling with Dirichlet smoothing is picked since not only does it offer the best ndcg@5, the performance is stable when its parameter μ is set greater than 1000 (less than 2% difference). Thus, we use it as the basic retrieval function throughout our experiments and set μ to be 2500. This baseline is called *AW* for the rest of the chapter as all words in the event article are used.

4.2.4 Time Filter

In this section, we investigate the temporal aspect of the background information. First, analysis is conducted to test our intuition that background information is more likely to be information published prior to the query article. In order to do that, the judgment data for the two years' of News Track is merged and, for each query article, the percentage of background documents published before it is computed. The percentages are grouped as *all*, *event*, *non-event* based on whether their corresponding query articles are event articles or not. The percentages of the groups are shown as box plots in Figure 4.3. As can be seen, the plot largely confirms our intuition. Half of the query articles, regardless of their types, have no less than 80% of their background articles published prior to them, and this percentage is around 65% for two-thirds of all the query articles. However, the tendency is more pronounced for non-event articles. All non-event query articles have more than 50% background articles published before the query articles. For event query articles, however, there are a handful of them that have more than 50% background articles written after. A possible explanation is that, in non-event articles such as an opinion piece or investigative reporting, the background tends to be the things that are commentated on or are investigated, which are more likely to be already published about in the past. For event articles, however, background can be new revelations of the event after the initial reporting, which are therefore published afterward.

Based on this observation, a time filter is applied on top of AW to ensure that all returned documents are published before the corresponding event articles. This method is named *TF*. The comparison of the average ndcg@5's on whether the filter is applied for different article types of two years' of News Track data is shown in Figure 4.4 with the black error bars indicating confidence intervals on the level of 95%. As can be seen, *TF* improves the performances for both event and non-event articles. With the help of the filter, the ndcg@5 increases from 0.4896 to 0.5159, and from 0.5002 to 0.5417 for event and non-event articles, respectively. The difference, however, is only statistically significant for non-event articles at the level of 95% according to the Wilcoxon signed-rank test. This is not surprising due to the fact that non-event articles tend to have more background articles published in the past. These improvements also transfer to statistically significant increases if both types are considered. The experiment illustrates that the time filter could be beneficial in practice for background information retrieval.

4.2.5 Aspect Based Background Retrieval for Event Articles

In this section, we investigate the usefulness of the proposed aspect based background retrieval framework, and that of its three components: entity graph based aspect identification, aspect language model estimation, and aspect importance estimation, which are the focus of this work.

Figure 4.3: The box plot of the percentages of background articles published earlier than the corresponding query articles for different article types

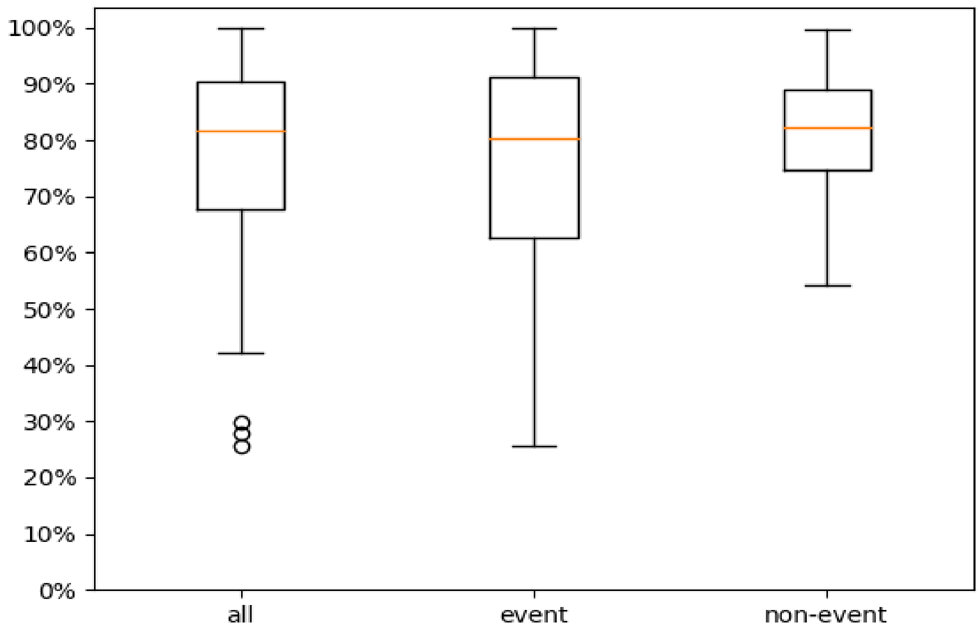
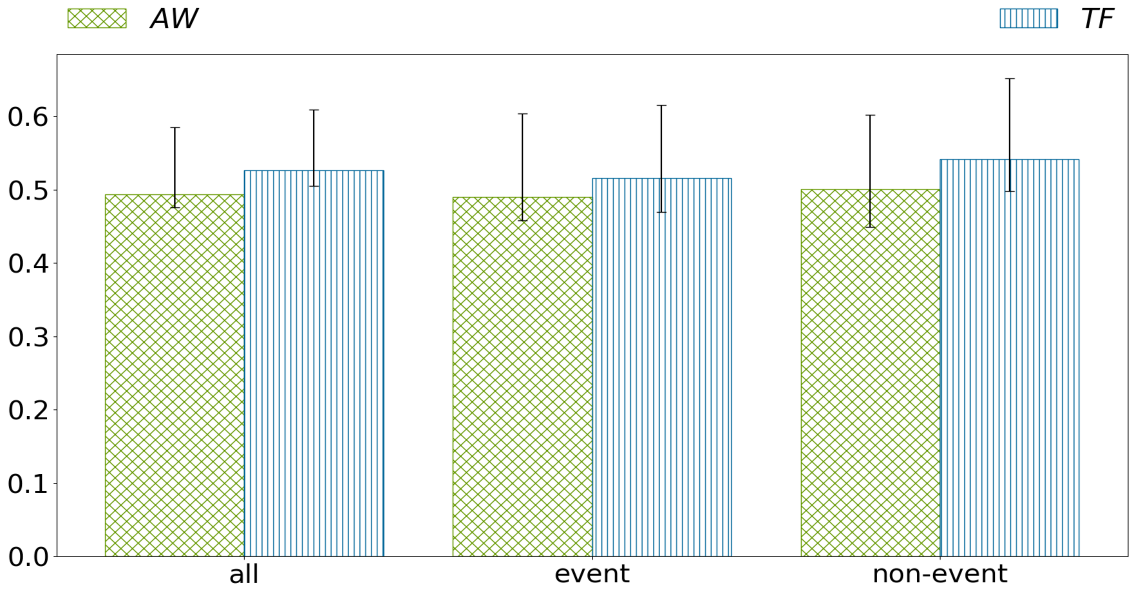


Figure 4.4: The comparison on the effect of the time filter for different article types



4.2.5.1 Framework Performance

In this set of experiments, we aim to test whether the proposed aspect based background retrieval framework can help background retrieval for event articles. As mentioned previously, it is employed in a re-ranking manner. The initial results to be re-ranked is produced by *TF* since it is slightly better than *AW* on event articles, and *AW* is used as a baseline for comparison.

Besides *AW*, in order to better understand the usefulness of the entity graph based aspect identification, a set of other baselines are also implemented which have different aspect estimation methods. More specifically, instead of using the entity graph based aspect identification to assign aspect labels to paragraphs, it is assumed that there are five aspects (this number is empirically chosen) for each event article and each paragraph of the article can be generated by all of these aspects. Language models of the aspects are then estimated accordingly. This results in two baselines: *PLSA* and *LDA*, which respectively use LDA or PLSA on all paragraphs.

The baselines, as well as variants of the proposed background retrieval framework with different implementations of the components, are applied to the *judged* event articles in the two years' of News Track data and their effectiveness is reported in Table 4.1 as ndcg@5. Only judged event articles are used here since the purpose of this set of experiments is to show the effect of the proposed method on articles that are indeed event related. Experiments on the query articles that are *detected* to be event articles by the event article detection method will be discussed later. The hyperparameters, such as λ_C of L-PLSA and PLSA that controls the influence of the collection model, and β in Equation 4.1 that balances the document aspect likelihood and document relevance are set by 5-fold cross-validation and the average effectiveness over the 5 folds are reported in the table. For the proposed method, different variants are implemented with the same aspect identification method but different combinations of aspect language model estimation methods (e.g. *L-PLSA* and *L-LDA*) and aspect importance estimation methods. For the latter, the average aspect weighting, clarity based aspect weighting, and co-occurrence based relatedness weighting are noted as *Avg*, *Clarity*,

Table 4.1: Effectiveness (ndcg@5) of different component combinations. † indicates statistically significant difference against the *TF* baseline with $p < 0.05$ according to the Wilcoxon signed-rank test.

TF		0.5178	
$PLSA + Avg$	0.5077	$LDA + Avg$	0.5116
$PLSA + Clarity$	0.4889	$LDA + Clarity$	N/A
$PLSA + Rela_{CO}$	0.5078	$LDA + Rela_{CO}$	0.5164
$L\text{-}PLSA + Avg$	0.5323	$L\text{-}LDA + Avg$	0.5132
$L\text{-}PLSA + Clarity$	0.5339	$L\text{-}LDA + Clarity$	N/A
$L\text{-}PLSA + Rela_{CO}$	0.5239	$L\text{-}LDA + Rela_{CO}$	0.5175
$L\text{-}PLSA + Rela_{wc}$	0.5233	$L\text{-}LDA + Rela_{wc}$	0.5191
$L\text{-}PLSA + Rela_{clo}$	0.5248	$L\text{-}LDA + Rela_{clo}$	0.5130
$L\text{-}PLSA + Rela_{bet}$	0.5159	$L\text{-}LDA + Rela_{bet}$	0.5177
$L\text{-}PLSA + Rela_{deg}$	0.5313	$L\text{-}LDA + Rela_{deg}$	0.5168
$L\text{-}PLSA + Rela_{eig}$	**0.5323**†	$L\text{-}LDA + Rela_{eig}$	0.5174
$L\text{-}PLSA + Rela_{Tex}$	**0.5338**†	$L\text{-}LDA + Rela_{Tex}$	0.5164

and $Rela_{CO}$. On the other hand, the graph based relatedness methods are noted by the node importance methods as follows: $Rela_{wc}$ for word count, $Rela_{clo}$ for closeness centrality, $Rela_{bet}$ for betweenness centrality, Rel_{eig} for eigenvector centrality, $Rela_{deg}$ for degree centrality, and Rel_{Tex} for TextRank. Since the aspect weighting method *Clarity* requires the percentage of the coverage of the aspects in paragraphs, which is a byproduct of *PLSA* and *L-PLSA*, it cannot be used in combination with *LDA* or *L-LDA*. Similarly, the entity graph based aspect weighting methods are not used with *PLSA* and *LDA* since, unlike their proposed counterparts, entity graphs are not used.

As can be seen in the table, the proposed framework can be effective if proper implementations of the components are chosen. Combining *L-PLSA* with either $Rela_{eig}$ or $Rela_{Tex}$ can bring statistically significant improvements over the *TF* baseline. However, baselines that leverage the proposed retrieval framework but do not employ aspect identification perform worse than *TF*. This might indicate that using entity graphs to identify aspects is helpful since it enables the language model estimation of the aspects to be performed in a more targeted fashion.

When comparing the proposed aspect language model estimation methods by looking at the table column-wise, it seems that *L-PLSA* provides more accurate estimation. Statistically significant improvements can only be achieved by using *L-PLSA*. Moreover, with the same aspect weighting methods, *L-PLSA* generally outperforms *L-LDA*.

On the other hand, when comparing the aspect weighting methods by looking at the table row-wise, there can be multiple observations. First of all, surprisingly, the simple *Avg* method achieves relatively high effectiveness among different aspect weighting methods if proposed language model estimation methods are in use. Per query analysis, nevertheless, indicates that it lacks robustness. When *Avg* is combined with *L-PLSA* and *L-LDA*, the number of queries whose performances are hurt is either close to or higher than the number of queries whose performances are boosted. This might be expected since *Avg* does not distinguish between aspects and assumes that they are equally important. A similar lack of robustness can be observed for *Clarity*. This is unsurprising as well since the underlying assumption of it can be violated moderately often. *Clarity* assumes that the less amount of text that is used for an aspect, the aspect is less likely to be described clearly, which results in a need for more information of the aspect. However, the reason why the authors do not cover the aspect in detail might be that the aspect is not important. The best aspect weighting method type seems to be relatedness based, especially those using entity graphs. However, some node weight methods seem to be advantageous than others. A possible explanation is that some of them, such as closeness and betweenness, can overly favor nodes that connect to aspects that contain more entities nodes than others. For instance, the closeness of a node is related to the shortest paths from it to other nodes in the graph. If a node of the event is well connected to an aspect with many nodes, the shortest paths of it to the nodes in the aspect might be smaller than that of other event nodes, which leads it to receive high closeness. This, in turn, would result in high importance to the aspect. However, the local structure of the graph has less impact on the measures such as eigenvector centrality and TextRank since they are built on the assumption that if a node is connected to important nodes, the node itself might be important, which has a weight smoothing effect.

4.2.5.2 Usefulness of Aspect Identification and Aspect Language Model Estimation

In this section, we analyze the effect of the entity graph based aspect identification and aspect language model estimation, which are two important components of the proposed news retrieval pipeline. The quality of the identified aspects can be measured by whether the entities assigned to each aspect form a meaningful theme as well as whether the aspect label assignment to the paragraphs correctly corresponds to the locations of the aspects in the event article. Likewise, the accuracy of aspect language models estimation can be measured by how accurate they represent the words used to describe the corresponding aspects. Both of the evaluation methods would ideally involve manual efforts. However, due to the limit of resources, we try to evaluate both at the same time by assessing the background retrieval effectiveness of using individual aspects. It is arguably true that the retrieval effectiveness with only one aspect reflects the quality of the aspect as well as its language model accurately.

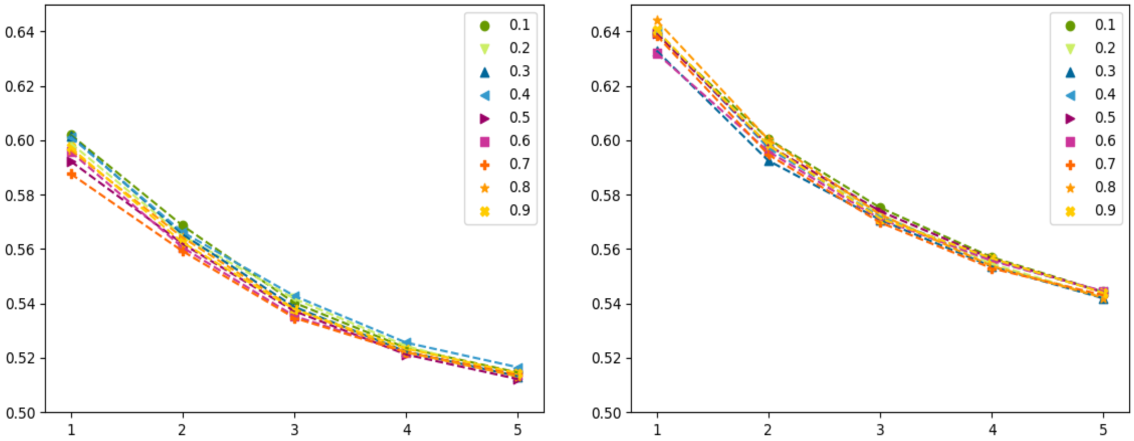
Based on the evaluation idea described above, we implement a procedure which consists of two steps: background retrieval for individual aspects and top aspects selection. More specifically, at the first step, for each aspect of an event article, a background article list is produced by re-ranking the results of *TF* using the framework specified in Equation 4.1 with only the language model of this aspect in use and the aspect importance $Pr(a_i|Q)$ discarded. After the first retrieval step, the top N aspects in terms of their background article lists' ndcg@5 are picked and these top ndcg@5's are averaged for comparison. Only a subset of aspects is picked since, given an event article, there may not be too many aspects that a reader is interested in. We vary N from one to five since this range seems to reflect the number of important aspects for a given article based on our observation. The weight regulator β between the aspect and the event article in the first step also depends on N. For each N, β is picked to be the one that offers the best average performance for all event articles. Optimal β

is chosen to limit the effect of the retrieval model on the retrieval performance so that the average ndcg@5 of top N aspects can serve as a more accurate proxy of the quality of the picked aspects.

Four methods are included in this set of experiments, which are called *L-PLSA*, *L-LDA*, *PLSA*, *LDA*. For *L-PLSA* and *L-LDA*, similar to the previous section, aspects are identified using the entity graph based method whereas the language model estimation is achieved by *L-PLSA* and *L-LDA*, respectively. Likewise, *PLSA* and *LDA*, which serve as baselines, set the number of aspects as five with no aspects labels, and the language models are estimated as their names indicate. The evaluation procedure described above is subsequently applied to the mined language models of these methods. Since *L-PLSA* and *PLSA* are different than the other two methods in that they contain a model parameter λ_C which controls the impact of collection language model (i.e. Equation 4.5). Thus, they are compared first with λ_C varies from 0.1 to 0.9 with a step size of 0.1. The average performances of top N aspects are shown in Figure 4.5. In the figure, the x-axis indicates the value of N, which is the number of top aspects, and the y-axis indicates the average ndcg@5. As can be seen, *L-PLSA* seems to be advantageous compared to *PLSA*. The former outperforms the latter for every λ_C at every N value. Moreover, *L-PLSA* seems to be a slightly more robust method with respect to λ_C as the variance in average performances at different N value tends to be smaller in *L-PLSA*, especially when N is small.

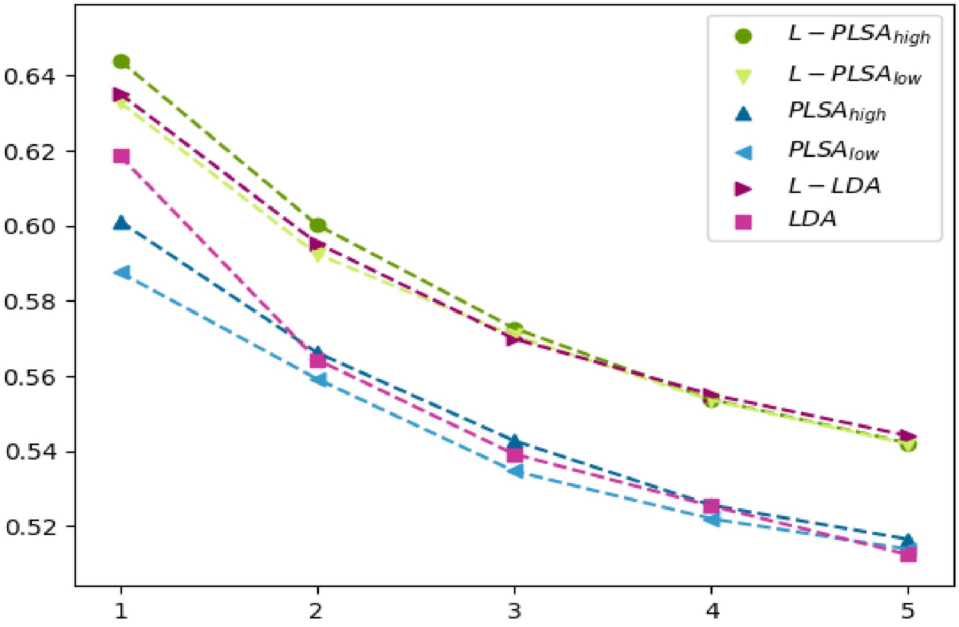
After these two methods using different variants of *PLSA* are compared, we also compare them against *L-LDA* and *LDA*. For simplicity and clarity, instead of showing the performances on all λ_C values, the best and worse performing λ_C's are chosen for *L-PLSA*, and *PLSA* and they are denoted using the subscripts *high* and *low*, respectively. The average ndcg@5 over different N's for them, as well as that of *L-LDA* and *LDA*, are shown in Figure 4.6. The most salient observation may be that the methods with the proposed aspect identification method, which are *L-PLSA*$_{high}$, *L-PLSA*$_{low}$ and *L-LDA*, consistently outperform the baselines for all N values. When comparing different

Figure 4.5: Average ndcg@5 with different λ_C for *PLSA* and *L-PLSA* at different N values


(a) *PLSA*

(b) *L-PLSA*

Figure 4.6: Average ndcg@5 for different aspect identification and language model estimation methods at different N values

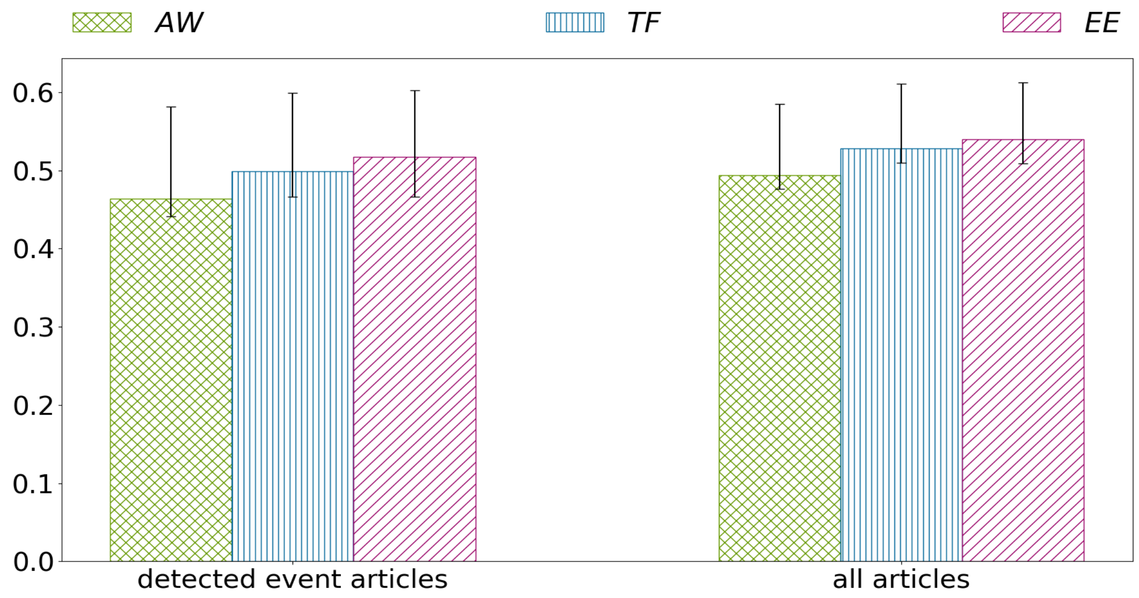


language model estimation methods among the proposed methods, it seems that *L-LDA* offers virtually the same performance as the worst *L-PLSA* setting, although as N increases, all proposed methods are virtually indistinguishable. This may suggest that *L-PLSA* could estimate language models more accurately. This is in accordance with the observation on the overall framework performance in Table 4.1, in which the best performing method uses *L-PLSA*.

4.2.6 End-to-End Pipeline Effectiveness

So far, the effect of different components of the proposed news retrieval pipeline is studied, such as the accuracy of the event article detection as well as the effectiveness of the time filter and the aspect based background retrieval model. Therefore, in this section, we investigate the end-to-end usefulness of the whole pipeline. That is, given a query article, it is classified as event or non-event via the event article detection method. If it is detected as an event article, the time filter as well as the aspect based background retrieval framework is applied. Otherwise, only the time filter is applied. It is important to note that in the aspect based background retrieval framework, the aspect language model estimation method and the aspect important estimation method are *L-PLSA* and Rel_{Tex}, respectively. This is due to the fact that this combination performs the best not only on the true event articles, but also in the end-to-end settings. We name this end-to-end method as *EE*. It is compared with baselines *AW* (which uses language model as the retrieval function and all words in the event article as the query) and *TF* (same as *AW* but no time filter applied) in terms of average ndcg@5 in Figure 4.7, and the confidence intervals at the level of 95% are shown as error bars. As can be seen, on the detected event articles, which is determined by the event article detection method, *EE* outperforms *AW* and *TF*, and the improvements are both statistically significant at the level of 95% according to the Wilcoxon signed-rank test. However, similar to the findings on true event articles in Figure 4.4, *TF* alone cannot bring significant improvement on *AW*. It might suggest that, due to the high accuracy of the event detection method, event articles are correctly identified and the performances on

Figure 4.7: The effectiveness of the end-to-end pipeline



them are increased with the help of primarily the aspect based background retrieval framework and also the time filter. In addition, although non-event articles is not a focus of the work, we also include them in the comparison combined with the event articles since the time filter also seems to be helpful on them, which is illustrated in Figure 4.4. Unsurprisingly, on all query articles, *EE* is also advantageous against the other two methods with statistically significant improvements.

4.3 Summary

In the chapter, we investigate the event background retrieval in the news retrieval pipeline of the proposed unified framework. We propose a simple rule based method to detect event article. Experiment results illustrate that it can accurately detect the event articles with high recall. Moreover, in terms of retrieving background information for events, a simple time filter on the publication time is introduced and an aspect based background retrieval framework is designed to leverage the mentioning of background information in the event articles to find more details of the mentioned background aspects. Experiment results show that the time filter provides small improvements for event articles, and significant improvements on non-event articles. Additionally, for the aspect based background retrieval framework, we experimentally show that it is effective in retrieving background information. Further analysis indicates the usefulness of its components. The entity graph based aspect identification seems to be able to accurately identify aspects and improve the aspect language model estimation accuracy by corresponding correctly the aspect language models to be estimated with paragraphs of the aspects. Moreover, some of the aspect importance estimation methods which rely on entity graphs, such as $Rela_{Tex}$, are shown to be effective. With the help of different components, the whole pipeline brings end-to-end statistically significant improvements against a competitive baseline on not only event articles, but also non-event ones.

Chapter 5
MICROBLOG RETRIEVAL

In this chapter, we discuss the microblog retrieval pipeline of the framework. First, we discuss how general microblog retrieval can be improved by leveraging the proposed query collectivity based measures on silent day detection and pseudo relevance feedback. How we plan to leverage these measures for these tasks are explained. Silent day detection is formalized as a classification problem, and the proposed measures are used as features. On the other hand, the measures can be used in pseudo relevance feedback to directly select feedback terms. Unlike traditional pseudo relevance feedback methods, which select terms from top retrieved documents based on the estimated relevance of the documents and the importance of the terms in the documents, the query collectivity based measures directly choose terms that, when combined with existing query terms, can lead to the increase the measures. Based on the design of the usages for the proposed measures on these two tasks, a general scoring function for the measure is proposed. According to the function, two realizations of it, which are phrase-based weighted information gain (PWIG), and local query term coherence (LQC), are discussed. Experiments are conducted to evaluate the usefulness of the measures on both tasks.

Besides only using microblogs, how the background aspects identified from news articles in the news retrieval pipeline can be leveraged to bridge news retrieval and microblog retrieval specifically for event queries are discussed. More specifically, to achieve this, we propose a search process in which event queries are searched on news and microblogs using a classic retrieval method. Following the procedure described in the previous chapter, aspects are identified from the result news articles. Their weights with respect to the articles as well as their language models are estimated.

The language models are then used alongside the original query to re-rank the results from the microblog side for the aspects. Additionally, in order to help prioritize more important aspects, we propose a weighting scheme to assign cross-article comparable weights to aspects based on the weights of the aspects obtained from the news retrieval pipeline and the estimated relevance of the articles. Top terms of the language models can also be provided to the users to help them choose the aspects they are interested in. An experiment to evaluate such a paradigm is conducted on a mixture collection consisting of a news collection and multiple tweet collections.

5.1 Query Collectivity Assisted Microblog Retrieval

5.1.1 Silent Day Detection

We first define the problem of silent day detection. Let Q denote a query and C denotes all microblog posts of a day. The random variable S represents whether the day is silent (1) or not (0). The problem of silent day detection is to estimate the following probability:

$$P(S = 1 \mid Q, C). \tag{5.1}$$

Because silent day detection requires a system to make a binary decision, it is formulated as a classification problem and the proposed query collectivity measures are used as features.

Based on the definition of silent day detection, it is natural to compute the measures on all microblog posts C. However, this might not be an effective choice. The differences between a silent and a non-silent day can be as small as a few, or even a single, relevant posts. Such differences might not lead to changes on collection wise measures that are significant enough to be detected. Instead, a possibly better solution is to use a retrieval function to retrieve a list of results first as a sampling method to obtain microblog posts that are more likely to be relevant, and compute the proposed measures on the list. If the retrieval function is able to retrieve some relevant posts for a non-silent day, the differences of the relevance measures between the retrieved lists of a silent and a non-silent day may certainly be easier to detect than that on all the posts of the days.

In addition, existing work also seems to suggest that compute the measures on the retrieved lists might be preferred. Missing content detection [59] is a similar problem to silent day detection that requires a system to detect the non-existence of relevant information on a *document* collection with respect to a query. The state-of-the-art method of it also leverages retrieval functions to obtain result lists first and bases the predictions on the lists. Another line of research that is also similar to silent day detection is query performance prediction [44, 12, 38, 2, 11, 62, 61, 46, 59, 10, 54], which estimates the difficulty to retrieve relevant results for a query. In fact, silent days can be deemed as an extreme case of query difficulty. There are two types of query performance prediction methods: pre-retrieval predictors and post-retrieval predictors. The difference is that the latter leverages some retrieval function to obtain a list of documents, and the prediction is conducted on the list with respect to the query. Pre-retrieval predictors, however, rely on collection wise measures. Previous research indicates that post-retrieval predictors generally provide better predictions over pre-retrieval predictors [62, 12, 11]. Based on the above two reasons, given a query and microblog posts of a day, we perform retrieval on all the posts of a day for the query to obtain a result list, and compute the proposed query collectivity measures on the list.

5.1.2 Pseudo Relevance Feedback

Another retrieval problem we try to improve by using the query collectivity measures is pseudo relevance feedback. More specifically, the proposed measures can be used as a way to directly select expansion terms. Following the assumption of pseudo relevance feedback, we assume top-ranked documents L_K of some retrieval function are relevant, and try to select expansion terms from these feedback documents. The selection criterion is whether adding a term to the original query can lead to an increase of the proposed measures, and how much is the increase. Operationally, given a query Q, we can first compute the query collectivity measures on L_K for Q. For each term t

that occurred in L_K but is not a query term, it is added to the query, and the measures for the expanded query is computed again on L_K. The term that leads to the highest increase of the measures is selected. This procedure is conducted iteratively until no terms can be found to further increase the measures.

5.1.3 Query Collectivity Measures

Based on the usages of query collectivity measures on silent day detection as well as pseudo relevance feedback, we formulate the general scoring function of query collectivity as follow:

$$S(T, L) = \sum_{C \in T} S(C, L), \tag{5.2}$$

where T is a set of terms and L is a list of documents. D_i is a document in L. C is a *group* of terms in T instead of a single term. As mentioned earlier, both tasks use the list of top-ranked results for the query. Therefore, L represents such a list. However, T represents different term groups for different tasks. For silent day detection, T represents all query terms since the objective is to measure the relevance of L for the query. For pseudo relevance feedback, on the other hand, initially the measures are computed for the query Q. Subsequently, T represents the expanded query in the iterative process which consists of a new non-query term t combined with either the original query Q, or the expanded query from the previous iteration. We propose two types of instantiations of the function, which are **phrase-based weighted information gain** (PWIG), and **local query term coherence** (LQC). We discuss them in detail below.

5.1.3.1 Phrased-Based Weighted Information Gain (PWIG).

The first measure is called phrase-based weighted information gain (PWIG). It infers relevance from the collective query term appearances. This measure is inspired by weighted information gain (WIG) [62], which is a query performance predictor that estimates the performance of a list of retrieved results for a query without the use of relevance judgments. It accomplishes the prediction by estimating the relevance of the retrieved results via summing up the relevance gains of single or collective query term appearances in the results. It has two components that measure the collective term appearances: sequential dependence (considering term order) and full dependence (without considering term order) features. The collection frequency is used to determine the weights of single and collective term appearances. We remove its single term appearance gains and adopt its multi-term appearance gains. Moreover, different weights are introduced for sequential and full dependency features. More specifically, given a query Q, we denote its sequential dependence feature set and full dependence feature set as $S(Q)$ and $F(Q)$, respectively. Let the union of $S(Q)$ and $F(Q)$ be noted as $R(Q)$. ξ denotes either a sequential dependence feature or a full dependence feature. The probabilities of ξ occurs in a document D and the whole collection C are denoted as $P(\xi|D)$ and $P(\xi|C)$. Given a list L containing K documents $D_i \in \{D_1, D_2, ..., D_K\}$, PWIG is computed as follow:

$$PWIG = \frac{1}{K} \sum_{D_i \in L} \sum_{\xi \in R(Q)} \lambda_\xi log \frac{P(\xi|D_i)}{P(\xi|C)}, \text{ if } |R(Q)| > 0, \tag{5.3}$$

where

$$\lambda_\xi = \begin{cases} \frac{\lambda}{\sqrt{|R(Q)|}}, & \xi \in F(Q) \\ \frac{1-\lambda}{\sqrt{|R(Q)|}}, & \xi \in S(Q) \end{cases}. \tag{5.4}$$

Essentially, PWIG infers relevance from the existence of multiple terms instead of single terms. For single term queries, WIG is computed instead.

5.1.3.2 Local Query Term Coherence (LQC).

Local query term coherence (LQC), however, measures how coherent the query terms are. The rationale behind it is that if the query terms are closely related (i.e. co-occur often) in a list of documents, it is more likely that the list is relevant. More specifically, the sub-coherences are computed, which are the coherences of the subsets of query terms. Sub-coherences are aggregated as the coherence of the query. More specifically, given a query Q, we denote all possible multi-term subsets of the query as $F(Q)$. We use the appearance of such a subset $\xi \in F(Q)$ as an indication of coherence among the query terms in ξ. The proportion of the documents containing ξ in a list of K documents indicates the degree of coherence between these query terms in the list:

$$C(\xi) = \frac{\text{\# of documents containing } \xi}{K}. \tag{5.5}$$

Intuitively, the appearance of more query terms collectively means a higher degree of coherence since the probability of more query terms randomly co-occurring is lower than that of fewer terms. Therefore, ξ are grouped by their sizes. More specifically, we denote all the query term subsets with size i as $F_i(Q)$ where i can take the value from 2 to the length of the query $|Q|$. Then the coherence $C_i(Q)$ aggregating all query term subsets with length i can be computed as:

$$C_i(Q) = A(\{C(\xi) : |\xi| = i\}). \tag{5.6}$$

The function $A()$ can be Max, Average, or Binary. Binary means that the value of $A()$ is 1 as long as one of the coherences is not zero, or 0 otherwise. This function is sensitive to the change from no subset appearances to one appearance. We use a weighted sum of the coherences of query term subsets of different lengths as the raw LQC ($rLQC$):

$$rLQC = \sum_{i=2}^{|Q|} \log(i) \times C_i(Q) \tag{5.7}$$

It is important to note that the weight of each size is the logarithm of the size. This way of assigning weights favors long query term subsets without overly penalizing the weights of short ones when the query is long. The final value of LQC normalizes $rLQC$ by the ideal LQC, which is the $rLQC$ of an ideal document list in which every document contains every query term.

5.1.4 Evaluation

In this section, we test how effective the proposed measures are on both silent day detection and query expansion on TREC collections.

5.1.4.1 Data

All the experiments for the proposed query collectivity measures are conducted over the same set of TREC collections. The set includes the collections used in TREC 2011 and 2012 Microblog Track [36, 48], TREC 2015 Microblog Track [25] and TREC 2016, 2017 Real-Time Summarization Track [26, 24]. These collections all consist of tweets crawled from Twitter, which a popular microblog platform. As a result, they are appropriate for testing pseudo relevance feedback on microblog retrieval. Moreover, in the latter three collections, silent days were introduced and evaluated. Although silent days were not discussed in the TREC 2011 and 2012 Microblog Track, they exist in the collection of the tracks. Therefore, it is suitable to evaluate silent day detection on them as well.

Tweets in all these tracks were crawled using Twitter streaming API. When active, it offers a 1% sample of all tweets posted at the time. The crawling period is 40 days in total, which resulted in 39,095,813 tweets. The total number of queries is 254. The queries in these collections were created by track organizers to reflect possible search interests related to the events that occurred during the crawling periods. We used the *title* field of the queries, which typically contains a handful of words. For each tweet, only text, tweet id, and timestamp were preserved and non-English tweets were discarded. Although experiments for both silent day detection and pseudo relevance feedback are conducted on the same set of tweet collections mentioned above, they are processed differently for each task, which will be discussed later.

5.1.4.2 Experiments for Silent Day Detection

5.1.4.2.1 Experiment Setup

For testing silent day detection, the tweet collections are grouped into three sub-collections based on the queries that they use and the details of them are described below.

- *TREC2011&2012*: This collection includes the data sets used in TREC Microblog Tracks of 2011 and 2012 [36, 48]. It was crawled during the period from January 24, 2011 00:00:00 UTC to February 8, 2011 23:59:59 UTC, which resulted in 4,833,223 tweets. 50 topics were used. Since the topics were timestamped, queries were not valid for all 16 days. There were 1344 query-day pairs, among which 355 were silent query-day pairs.

- *TREC2015&2016*: This collection includes the data sets used in TREC 2015 Microblog track and TREC 2016 Real-Time Summarization track [25, 26]. These two similar tracks require participating systems to post relevant tweets according to users' interest profiles in real time or at the end of each day. Data from these two tracks were merged together since there is query overlap. This collection was crawled in the same way as TREC2011&2012. There were two crawling periods that were from July 20, 2015 00:00:00 UTC to July 29, 2016 23:59:59 UTC, and from August 2, 2016 00:00:00 UTC to August 11, 2016 23:59:59 UTC. In total 25,968,078 tweets were crawled. There were 51 topics used in 2015 and 56 used in 2016, which resulted in 1070 query-day pairs, among them 257 were silent query-day pairs.

- *TREC2017*: This collection is the data set used in TREC 2017 Real-Time Summarization Track [24]. The problem setting, data collecting process, and query format are the same as that of the TREC2015&2016. The crawling period is from July 29, 2017 00:00:00 UTC to August 5, 2017 23:59:59 UTC. In total, there were 8,294,512 tweets and 97 topics, which resulted in 776 query-day pairs. Among them, 137 were silent query-day pairs.

In total, there are 3,190 query-day pairs including 749 silent query-day pairs, and tweets from the same day were grouped into a single-day corpus.

For retrieving the initial results list for the proposed measures to be computed on, we use the axiomatic retrieval method f2exp [16] as the baseline retrieval method in the experiments. It is important to note that any other retrieval functions can also be applied. For silent day detection, we use Naive Bayes as the classification method. In our preliminary study, we have tried other classification methods such as SVM and logistic regression and found that they perform worse than the Naive Bayes methods. A possible explanation is that our data set is skewed (only about 20% of the samples are positive), methods robust to unbalanced data, such as Naive Bayes, tend to outperform those that are not.

In order to illustrate the utility of the query collectivity based measures on silent day detection, we conduct the following experiments. First, we directly measure how effective the proposed methods are in detecting silent days by using them as features for the classification method. Afterward, the detectors were applied to the e-mail digest scenario of Microblog Track and Real-Time Summarization (RTS) Track to illustrate its usefulness for the real life application when recognizing silent days is important. Finally, we also evaluate the proposed methods on missing content detection [59] a similar problem but in a different domain.

5.1.4.2.2 Classification of Silent Days

We first implemented several baselines to compare our methods against. We first implemented a simple score threshold based method ST which decides whether a day is a silent day to a query by checking the highest score of the tweets of the day for the query against a score threshold. If the score is lower than the threshold, the day is classified as a silent day. This method is similar to those implemented by some top-ranked participants of TREC Microblog and RTS tracks to indirectly address the silent day problem by using per-tweet score threshold [64]. In addition, due to the similarity between silent day detection and query performance prediction, three query performance prediction based baselines, which are QPP_d, QPP_m, and QPP_c, are built. QPP_d consists of 12 state-of-the-art query performance predictors for *document* retrieval, such as WIG [62]. Whereas QPP_m employs 8 state-of-the-art query performance predictors for *microblog* retrieval, such as query term coverage and top term coverage [44]. The complete lists of features of these methods can be found in Table 5.1. QPP_c uses the *combination* of the predictors (20 predictors in total) in QPP_d and QPP_m. Additionally, we implemented a baseline $Tree$ which is a decision tree based method for missing content detection [59]. Missing content detection tries to detect queries with no relevant information in a document collection. It is a problem that is very similar to silent day detection and the difference is the type of collections. However, we did not incorporate query performance predictors into *Tree*, or the other way around, due to the unique feature format of *Tree*. *Tree* is based on the intuition that, for a query, if all the query terms rank similar sets of documents to be most relevant, the query might be "easy" and more likely to have some relevant documents. More specifically, it breaks each query into single-term subqueries. The subqueries are searched against the collection. The IDF of the query term in the subqueries, as well as the overlap between the search results of the subqueries and that of the original queries, are computed to form score tuples of *(single query term IDF, the number of overlap)*. These tuples are sorted based on the IDF values and are used to train a decision tree model. Starting from the root, the decision tree takes a tuple at a time based on the IDF order at each node, and the multiplication of the two items in the tuple is checked against a threshold associated with the node to decide which branch to take. Not only the difference in feature forms between $Tree$ and query performance predictors (a score tuple vs. a single value) but also the specialized decision tree for the $Tree$ method leads to our decision of not incorporating one method into another.

Proposed query collectivity measures, on the other hand, were incorporated into two methods TR and $TR + QPP_c$. In *TR*, only PWIG and LQC were used. $TR + QPP_c$, however, used the combination of TR and QPP_c. For LQC features, all three aggregation functions Max, Average, and Binary were used. It is important to note that, for a fair comparison, all features were tuned with the area under the curve (AUC) of the receiver operating characteristic (ROC) curve for optimal parameter settings.

In order to test how useful the proposed measures are for silent day detection, we designed two experiments. First, we performed 10-fold cross-validation for all methods in which all query-day pairs from the three tweet collections built for silent day detection were merged and divided into 10 equal size folds. We used 10-fold to avoid

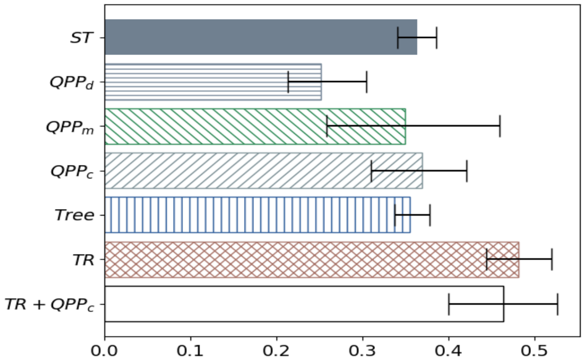
Table 5.1: Features of different baselines

Method	Features
QPP_d	average inverse document frequencies [2]
	simplified clarity [61]
	sum of variances of the term weights of top-k documents [61]
	maximum and average term relatedness [20]
	query clarity [10]
	standard deviation of top-k scores [38, 12]
	normalized standard deviation of top-k scores [12]
	normalized query commitment [46]
	weighted information gain [62]
	query feedback [62]
	top score of the retrieved list [54]
QPP_m	average, median, upper and lower percentile of query term coverage [44]
	average, median, upper and lower percentile of top term coverage [44]

bias and to ensure the number of folds is enough for meaningful tests of statistical significance. In every fold, there are 319 query-day pairs. This experiment tested the methods' performances on predicting silent days for *old* queries that the methods had seen and were trained on. The F1 score was used as the performance metric and the results are shown in Figure 5.1. In addition, the confidence intervals at the confidence level of 95% using the Wilcoxon signed-rank test are shown in the form of the black error bars on top of the bar for F1 scores for each method. As can be seen, when comparing more sophisticated baselines, QPP_m, QPP_c, and $Tree$ all outperform QPP_d. This is not surprising since QPP_m and QPP_c contain features dedicated to microblog retrieval. Moreover, $Tree$ was designed for detecting the non-existence of relevant information. However, further investigation shows no statistically significant differences between QPP_m, QPP_c, and $Tree$ at the 95% confidence level according to the Wilcoxon signed-rank test. When comparing all baselines, an interesting observation is that even the simple score threshold baseline ST performs similarly to the best baseline QPP_c with no statistically significant difference.

The effectiveness of the proposed query collectivity measures on silent day detection may be proven by this experiment. Methods containing them, such as TR and $TR+QPP_c$, outperform all baselines considerably with an improvement over the best baseline QPP_c more than 25%. The advantages of TR and $TR+QPP_c$ are also statistically significant. However, the difference between TR and $TR+QPP_c$ is not. It is

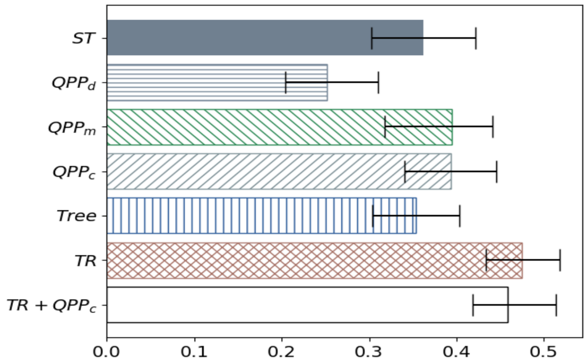
Figure 5.1: 10-fold cross-validation performances for different methods



interesting that, despite being designed to detect the absence of relevant information, $Tree$ performances significantly worse than TR and $TR+QPP_c$. A possible explanation is that $Tree$ uses the number of overlapping documents of top results between the original query and its single-term subqueries. Given the short nature of tweets, term document frequency is usually one for any term. Therefore, the ranking of the tweets for a single term subquery is likely to be dominated by the lengths of the tweets, which may not reflect the relevance order. As a result, the count of overlapping tweets of top results subsequently may not reflect the agreement on relevant tweets, which violates the underlying assumption of the method.

In the second experiment, we aimed to test the proposed detectors on the case of predicting silent days for *unseen* queries. We believe such a scenario occurs very often in real life. As the world constantly changes, new search interests are developed by the users. In this experiment, we conducted a modified version of 10-fold cross-validation used in the previous experiment where we divided *queries* into 10 equal size folds so that there is no query overlap among them. The performances as F1 scores are reported in Figure 5.2. A similar situation as the first experiment is observed. QPP_m, QPP_c, and $Tree$ are all better than QPP_d. ST also performs as well as these better baselines. On the other hand, TR and $TR+QPP_c$ are still the best-performing methods, and the difference between them and the baselines are statistically insignificant. It is interesting that in both experiments, adding QPP_c to TR hurts

Figure 5.2: Performances for testing on unseen queries



the performances. We computed the mutual information with the nearest-neighbor method between the features in QPP_c and day labels. Low mutual information is observed for several features with some close to 0. Therefore, adding these features might not be desirable. Based on the two experiments of testing proposed features, it can be concluded that the proposed query collectivity measures exhibit superiority over existing query performance prediction methods and the missing content detection method. For future applications that require silent day detection, it is advised to use TR.

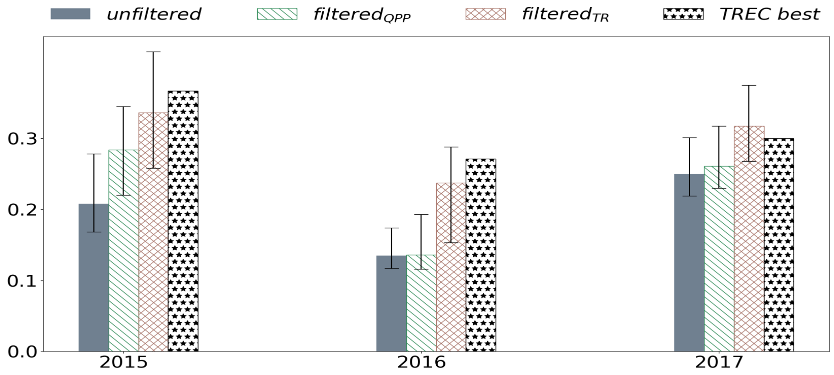
5.1.4.2.3 Effectiveness on Tweet E-mail Digest

Besides silent day classification, another way to examine the effectiveness of the proposed measures is to incorporate the TR silent day detector for the tweet e-mail digest scenario of TREC 2015 Microblog Track [25], and TREC 2016, 2017 Real-Time Summarization Track [26, 24]. In these tracks, a system's ability to detect silent days can substantially impact its performance [52]. A participating system is supposed to submit up to 10 tweets for a query-day pair if there are relevant tweets in that day, or 0 otherwise. The main evaluation metric is *ndcg@10-1* for 2015 and 2016 [25, 26] which rewards systems that can "keep silent" for silent days by the perfect score (1) for the days of the query. Submitting any tweets in a silent day would result in a score of 0. In non-silent days, ndcg@10 is computed instead. In 2017, however, the main evaluation metric is *ndcg@10-p*, which penalizes systems according to the number of returned tweets on silent days. In our experiment, we used *ndcg@10-1* for all three years since we investigate silent day detection as a classification task. It is possible to choose the number of tweets to return based on its probability of being silent as it would be more suitable for *ndcg@10-p*. We plan to explore this direction in the future.

We built a baseline method called *unfiltered*, which retrieves 10 tweets for every query on every day using f2exp. Another baseline method, $filtered_{QPP}$ was implemented by applying QPP_m on *unfiltered's* output to filter out silent days and return no results for them. We chose QPP_m since it is the best baseline when testing on new queries. It is important to note that QPP_m used for one year was trained on the collection we created excluding the queries of the year. The last method is called $filtered_{TR}$, which is very similar to $filtered_{QPP}$. The only difference is that the feature set TR was used instead of QPP_m. The performances of these methods as well as that of the best participating runs (e.g. $TREC_best$) for each year in terms of ndcg@10-1 [25, 26, 24] are reported in Figure 5.3. The confidence intervals at 95% according to the Wilcoxon signed-rank test are shown as the back error bars. It is important to note that confidence intervals are not reported for $TREC_best$ since we can only obtain the performance numbers from the TREC overview papers and do not have access to the actual run files. As can be seen, with the help of *TR*, $filtered_{TR}$ outperforms *unfiltered* and $filtered_{QPP}$ for all year, although the increases are statistically significant on 2016 and 2017 but not on 2015. When compared with TREC's best runs, $filtered_{TR}$ can offer comparative performances for 2015 and 2016. In 2017, it even outperforms the TREC best run. It is important to note that we only used the generic f2exp as the retrieval method. Nevertheless, some of the best runs' retrieval methods were further tinkered for microblog retrieval [64, 53] [1]. For instance, techniques such as query expansion with external resources and word embedding were used in 2015's best run [64]. Due to the consistently promising results of our method, it can be concluded that

[1] We did not find the TREC report of the best run of 2016

Figure 5.3: ndcg@10-1 of e-mail digest scenario of microblog tracks



the query collectivity measures are very useful and could enhance the effectiveness of real-world microblog retrieval systems significantly if silent days need to be addressed.

5.1.4.2.4 Additional Analysis

Since the proposed silent day predictors seem to be useful in the microblog retrieval domain, we also want to test whether they can be generalized to the document retrieval domain. Therefore, we examined them for missing content detection [59]. As discussed earlier, this problem is very similar to silent day detection, and the major difference is that a document collection is used.

The TREC collection Disk4&5, used for missing content detection in the previous research [59] was used. The *Tree* and *TR* method mentioned in the earlier section were tested on this collection. The performances are reported as F1 scores in Figure 5.4. As can be seen, *TR* is significantly worse than *Tree.* A potential explanation is that for the LQC features in *TR*, the distance between query terms in a document is ignored. This means that in the same document, no matter how much text there is between query terms, these terms are considered as related with respect to the document by LQC. This approach seems to be not always appropriate for document collection. However, it might be suitable for tweet collections due to the 140 character limit. To confirm this explanation, we modified the LQC features by adding size constraints when counting related query terms and applied the modified TR method, which is

Figure 5.4: Performances (F1) of missing content detection on Disk4&5

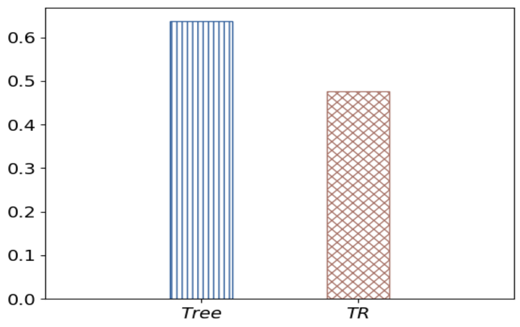


Table 5.2: Effect of size constraints on LQC features

Method	**Microblog**	**Disk4&5**
TR	0.501	0.476
STR	0.494	**0.606**

called STR, for both missing content detection and silent day detection. The results are shown in Table 5.2. We observed that, with the addition of the size constraints, the performance of *TR* can be boosted significantly (from 0.476 to 0.606) for missing content detection. Moreover, the size constraints do not seem to affect TR's performance on silent day detection. These observations confirm our assumptions.

The results of the experiments in this section as well as that of earlier sections about *Tree* on silent day detection suggest that applying methods for the task of detecting the non-existence of relevant information in a cross-domain fashion may not be appropriate. Not only does *Tree* fail in silent day detection, but also *TR* without size constraints tends to be ineffective in missing content detection. This discovery indirectly shows the value of our proposed measures since, to the best of our knowledge, they are the first measures designed for silent day detection for microblog retrieval.

5.1.4.3 Pseudo Relevance Feedback

5.1.4.3.1 Experiment Setup

In addition to silent day detection, we also conduct experiments to evaluate the effectiveness of the proposed query collectivity measures for pseudo relevance feedback. However, instead of grouping the tweet collections based on whether there is a query overlap, a setting similar to classic ad-hoc retrieval is used where year corpora are built by grouping all tweets for the same year, which results in 4 year corpora: 2011, 2015, 2016, and 2017. Queries of tracks of different years are searched on their corresponding year corpora.

Similar to silent day detection, we use f2exp [16] as the baseline retrieval method. After retrieving the first round of retrieval results, we apply the proposed measures to select expansion terms. There are two types of proposed methods: $PWIG$ and LQC. LQC has three score aggregation functions $Binary$, Max, and $Average$ as described before. This results in four different query collectivity based feedback methods which we denote as $PWIG$, LQC_b, LQC_m, LQC_a. For all of these methods, we employ a greedy algorithm to generate expanded queries. More specifically, we incrementally grow the original query by selecting one expansion term at a time that maximizes the increase of the corresponding relevance measure for the expanded query. The algorithm stops when there is no term that can increase the measure. In the expanded queries, original query terms and expansion terms are given different weights to regulate the impact of them on the retrieval results. Following the convention used in query expansion methods, we introduced a parameter β to balance the term weighting between original query terms and expansion terms. The weights of the original query terms are set to β and that of the expansion terms are set to $1 - \beta$ The proposed pseudo relevance method is compared with two baseline methods: f2exp [16] (no feedback) and RM3 [1], a state-of-the-art pseudo feedback method. RM3 estimates a relevance model from the top-ranked documents from an initial ranked list, top terms are then selected from the relevance model to expand the original query. More details can be found in Section 2.3.2 in related work. The performance metric ndcg@10 is used for evaluation following TREC setup [25, 26, 24].

Regarding the parameter setting, the parameters in the basic retrieval function is fixed to the same values used in previous subsections. For both RM3 and the proposed methods, the number of feedback documents is set to 10 and they differ on the number of feedback terms. This number is set as 10 for RM3. For the proposed collectivity based expansion methods, however, we do not need to specify the number of expansion terms. The methods' iterative expansion term selection will automatically stop if no additional terms can increase the measures. The parameter of β is tuned, and the best performance is reported unless otherwise stated.

5.1.4.3.2 Retrieval Effectiveness

Compared with state of art pseudo relevance feedback methods, one unique characteristic of our proposed method lies in its strict requirement in selecting terms. Most existing methods generate a long list of ranked terms and select only top terms as expansion terms. The number of expansion terms is often set as a fixed parameter for all queries. As a result, these methods might hurt the performance for some queries in which their top-ranked terms are not as high quality as in other queries. In fact, for those queries, it might be better not to expand any terms, but existing pseudo relevance feedback methods often are unable to do so since the parameters are set to the same values for all queries. On the contrary, our methods are much more conservative and restrictive in term selection due to the usage of query collectivity measures, and an expansion term is selected only when it can increase the query collectivity measures. As a result, our methods are able to not select any expansion terms for some queries. In fact, when they are able to select expansion terms, the number of expansion terms is often one.

Based on the discussion above, in the first set of experiments, we examine when the proposed methods can find expansion terms, whether the terms are effective in improving the performance. Table 5.3 shows the performance improvement of the proposed methods with respect to the baseline method (i.e., f2exp). Please note that

Table 5.3: Performance improvements over queries where the proposed methods can select expansion terms.

Measures used for term selection	**Performance Improvement**
$PWIG$	+1.6%
$\mathbf{LQC_b}$	**+15.6%**
LQC_m	+6.5%
LQC_a	+4.9%

Table 5.4: Performance comparison over all queries when only one term is expanded

Methods	**2011**	**2015**	**2016**	**2017**
$f2exp$	0.303	0.387	0.164	0.344
$RM3_1$	0.305	0.390	0.167	0.342
LQC_b	0.312	0.404	0.163	0.346

the performance is computed over only a subset of queries where our methods are able to select expansion terms. Although we find out that our methods on average can find expansion terms for only 10% of queries, the selected expansion terms are clearly useful. Among all the measures, LQC_b gives the best performance with the performance improvement close to 16%. This improvement is very encouraging since the it comes from only one expansion term. In the rest of the paper, we will focus on the LQC_b method.

The second set of experiments is to compare the effectiveness of our method with $RM3_1$, which uses the state of the art feedback method RM3 with the number of expansion terms set to 1. Since the proposed method selects at most one term, we want to see whether the quality of the selected expansion terms is better than the top one selected by RM3. Table 5.4 shows the performance comparison. It is clear that our proposed method can outperform $RM3_1$ over 3 out of 4 collections, indicating the effectiveness in selecting the most useful expansion terms. Moreover, we also examined the expansion terms selected by these two methods manually. For some queries, both methods select the same expansion term. For example, for query "political campaigns and social media", both methods select "American" as an expansion term. In other queries, such as “health insurance for disabled children”, the proposed method is able to select more useful expansion terms (i.e., “afford”) than the one selected by RM3 (i.e., “individual”). As a result, LQC_b has its ndcg@10 as 0.469 for this query whereas that of $RM3_1$ is only 0.290.

As can be seen so far, the advantage of the proposed method is its ability to select more useful expansion terms if it can find any, and its disadvantage is that it cannot find any expansion terms for many queries. In order to better utilize its advantage, we propose to combine it with existing query expansion methods such as RM3. Please note that we use $RM3_{10}$ (i.e. RM3 with 10 expansion terms) instead of $RM3_1$ because it leads to better performance.

The simplest combination strategy is to use the proposed method when it can find expansion terms and RM3 otherwise. This strategy is denoted as $Combine_{LQC}$. Intuitively, this combination can only be effective if the two methods LQC_b and $RM3_{10}$ are complementary to each other in terms of the subset of the query that they perform well. We conducted some analysis and it turns out that this assumption holds. In the analysis, all queries are partitioned into two groups based on whether LQC_b can find expansion terms or not. The average performance differences of LQC_b and $RM3_{10}$ against f2exp in both groups are reported in Figure 5.5. Based on the figure, we can see that when there are expansion terms from LQC_b, using these terms can be significantly better than using those from $RM3_{10}$. Moreover, $RM3_{10}$ seems to be able to boost the performance against f2exp when LQC_b cannot expand the query.

An alternative combination strategy is based on the query length. The strategy is proposed based on our observation that the query length is related to the performance improvement of these two methods. In particular, we compare the average performance differences of $RM3_{10}$ and LQC_b against f2exp across various query lengths and depict them in Figure 5.6. As can be seen, when there are less than or equal to 3 query terms, LQC_b is generally better than $RM3_{10}$. Conversely, when there are more than 3 query terms, $RM3_{10}$ performs the better. We closely examined the expansion terms of LQC_b which revealed that when the query is long (e.g. has more than three terms)

Figure 5.5: Improvements over f2exp for the two groups of queries

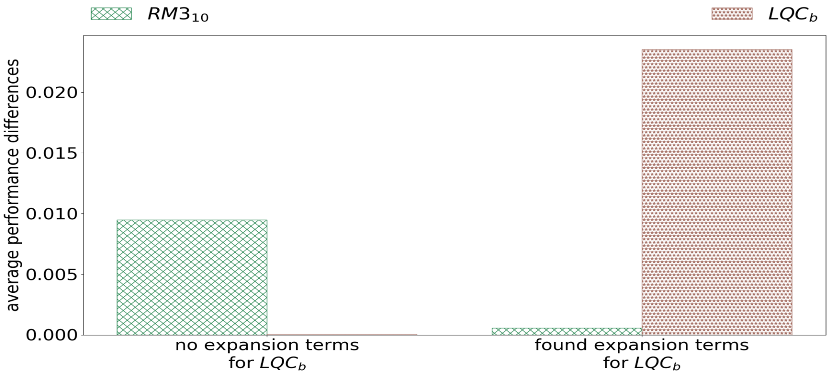
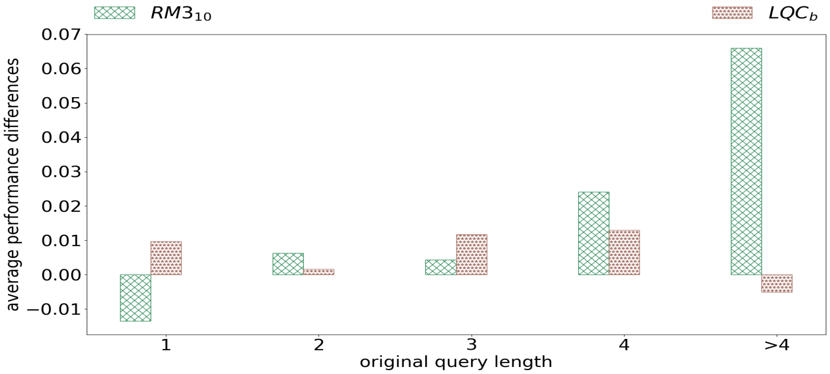


Figure 5.6: Performance differences between query expansion methods and f2exp across different query lengths



it is harder for query collectivity to guarantee that the expansion terms are relevant to the whole meaning of the query instead of only a partial aspect of it. For instance, there is a query "Johns Hopkins Lyme disease study" which contains two important aspects: Johns Hopkins hospital and Lyme disease study. However, LQC_b selected the term "Dr" which is the doctor title. It is closely related to "Johns Hopkins" but not to "Lyme disease study". Therefore, the result tweets are steered towards Johns Hopkins hospital, which results in the degradation of the performance. Based on the above discussion, we propose a second combination method denoted as $Combine_{LN}$, which uses the query length as a filter. If the length is greater than 3, $RM3_{10}$ is used. Otherwise, LQC_b is used.

The average performances among for these two new methods as well as that

Table 5.5: Compare different ways of combining LQC_b and $RM3_{10}$

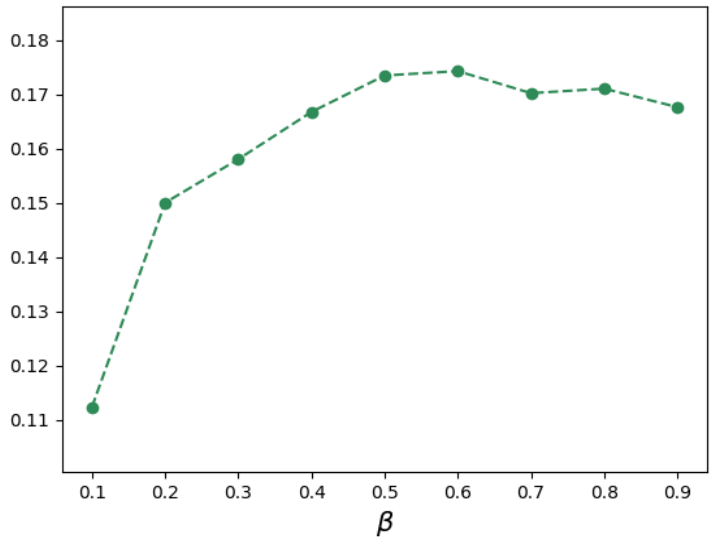
Methods	**ndcg@10**
$f2exp$	0.304
$RM3_{10}$	0.310
LQC_b	0.310
$\mathbf{Combine_{LQC}}$	**0.314†**
$\mathbf{Combine_{LN}}$	**0.317†**

of LQC_b, $RM3_{10}$, and f2exp are shown in Table 5.5 in which † indicates statistical significance according to the Wilcoxon signed-rank test with $p < 0.05$. The results are based on all the queries over all collections. It can be seen that although $RM3_{10}$ and LQC_b have higher ndcg@10 than f2exp, the differences are not statistically significant. However, both ways of combining them illustrate statistically significant gain over f2exp. Therefore, it can be concluded that although either LQC_b or classic query expansion method $RM3_{10}$ alone does not exhibit strong advantages over the non-feedback method f2exp, they can be complementary to each other and show significant improvement when working in concert.

5.1.4.3.3 Additional Analysis

Besides the above two sets of experiments, additional analysis is conducted to gain more insights of using query collectivity measures for pseudo relevance feedback. First, we investigate the impact of the parameter β, which regulates the weights between the original query terms and expansion query terms. It is tuned from 0.1 to 0.9 with a step size of one, and the average ndcg@10 across the queries that LQC_b can find expansion terms are shown in Figure 5.7. As can be seen, the method is relatively stable when β is higher than 0.5 in which cases the ndcg@10's are around the best value of 0.17. However, ndcg@10 decreases sharply when β is low and close to 0.1. The unstable performances on the lower end might not be an issue since setting β with such a low weight would lead to the expansion terms having dominantly larger influence on the retrieval over the original term, which seems to be often unreasonable.

Figure 5.7: Average ndcg@10 for different values of β for LQC_b



In addition to parameter sensitivity, we also investigate whether LQC_b, a pseudo relevance feedback method that seems to be useful for microblog retrieval, can be generalized to document retrieval. More specifically, LQC_b is tested on the Disk4&5 document collection. It is important to note that we used Mean Average Precision at 1000 (map@1000) instead of ndcg@10 as the performance measure for Disk4&5 because it is usually the metric for document collections and Disk4&5 in particular [1]. Our experiment results illustrate that LQC_b does not seem to boost the performance of f2exp as they could for microblog retrieval. This might be due to the fact that the initial retrieval results for document retrieval are not as noisy as they are for microblog retrieval and the assumption that these results are relevant is more likely to be held. Thus, a more aggressive expansion method such as $RM3$ can bring significant improvement since many useful terms are included. This can also be proven by our experiment results that $RM3_{10}$, which selects more expansion terms, outperforms $RM3_1$. LQC_b, however, is designed to be conservative to combat the noisy initial retrieval results on tweets and it only has expansion terms for 17 out of 250 queries for Disk4&5. As a result, it is unsurprising that its performance does not seem to be advantageous against baselines. Nonetheless, it would be interesting to see how it performs on difficult document collections where basic retrieval methods such as f2exp can hardly find any relevant documents. We leave this to future work.

5.2 Bridging News and Microblog Retrieval Using Background Aspects for Event Queries

In this section, instead of focusing solely on microblog retrieval, we propose to leverage the background aspects mined from event articles in the news retrieval pipeline to bridge news and microblog retrieval for event queries. That is, if the same event is searched on both news and microblogs, the background aspects identified in news are used to re-rank the retrieval results on the microblogs. This method not only provides the possibility for the users to explore the microblog posts with finer granularity, but also can complement the background information from news, which is fact-based, with often opinionated microblog posts.

5.2.1 Methodology

For re-ranking microblog posts of a query Q for an aspect a_i that is identified from news, following the formula 4.1 for ranking background articles for aspects, a microblog post T is assigned with a score using the equation below:

$$S\left(T \mid a_i, Q\right) = \gamma S\left(T|Q\right) + (1-\gamma) S(T|a_i), \tag{5.8}$$

in which both $S\left(T|Q\right)$ and $S(T|a_i)$ can be computed using existing ad-hoc retrieval techniques with T and Q being represented by their text and a_i being represented by its estimated language model. γ balances the weight between the original query and the aspect.

We envision that microblog posts are re-ranked using the above model in a uniform way, but are presented differently for the two sources. On the news side, similar to the background articles, the microblog posts re-ranked for an aspect can be shown in the sidebar alongside the corresponding aspects. Alternatively, instead of showing the posts by default, they can be hidden initially and users are given the option to show them if they are interested in the aspects. By rendering microblog posts physically close to the corresponding aspects, when users read an aspect that they are interested in, relevant microblog posts can be obtained with minimum efforts.

On the microblog side, however, the re-ranked microblog posts are shown after the initial retrieval results for the query are presented. More specifically, aspects are ranked based on their importance to the query, and text descriptions of the aspects are shown to the users. Users then can pick the aspects for which the microblog posts are re-ranked and shown. These two components might be required for the method to be useful since they both can assist users in choosing the aspects to explore. Ranking aspects can promote those that users are more likely to be interested in at the top, whereas presenting text descriptions seems to be necessary since users need a way to understand the aspects before picking those that they want to explore.

The presentation of the microblog posts for news articles does not require extra processing besides microblog post re-ranking. On the microblog side, however, the importance of aspects needs to be estimated and the text descriptions need to be generated. Although in the news retrieval pipeline section, the methods for estimating aspect importance within an article are discussed, since there can be multiple articles related to the event, additional processing steps are needed to normalize the aspect importance so that they are comparable across articles. To accomplish this, a method is proposed as shown below:

$$Pr\left(a_i \mid Q\right) = Pr\left(a_i \mid Q, D\right) \frac{Pr(Q \mid D)}{\sum_{D' \in \mathcal{R}} Pr\left(Q \mid D'\right)}. \tag{5.9}$$

In the above equation, Q represents an event related query whereas R represents a set of event article retrieved for Q. D denotes a document in R, and a_i denotes an aspect in D. $Pr\left(a_i \mid Q, D\right)$ represents the weight of the aspect in the document with respect to the query, which is the output of the aspect importance estimation described in the news retrieval pipeline. On the other hand, $Pr(Q \mid D)$ is the relevance probability of the document to the query, which can be approximated by the scores for the document of the underlying retrieval function. This weighting method is built on the intuition that if an aspect is important to a document with respect to the query, and the document is likely relevant to the query, the aspect is likely to be important to the query.

Besides the cross-article aspect importance, another critical part of the method is the text descriptions of aspects. We follow the previous work [30] and choose to use top terms from the language models of the aspects since it could help users to establish a reasonable understanding of the aspects.

5.2.2 Data and Experiment Setup

To the best of our knowledge, there are no collections appropriate for testing the effectiveness of using background aspects in news to guide microblog retrieval. Thus, we try to use existing news and microblog collections to design an experiment that the proposed method can be appropriately tested. It is natural to reuse the tweet collections used for silent day detection and query expansion previously, which include the tweet collections of TREC 2011 and 2012 Microblog Track [36, 48], TREC 2015 Microblog Track [25] and TREC 2016, 2017 Real-Time Summarization Track. For news collection, the Washington Post collection used in the news retrieval pipeline is selected. However, since they are built separately for different purposes, proper data processing needs to conducted so that the data are suitable for the experiment. Moreover, based on the data, the experiment needs to be properly designed to appropriately test the proposed method.

5.2.2.1 Data Processing

At the first step of processing the data, we focus on the *time alignment* between the queries and the two collections. Since the proposed method is designed for event related keyword queries, which exist in the tweet collections but not in the Washington Post collection, queries from the tweet collections are used. However, searching them freely on the collections without any time restrictions may not be appropriate. Instead, it is essential to ensure that the queries are searched on tweets and news published around the time the event happened. Intuitively, given an event, it is likely that the discussions about it on Twitter as well as the news coverage for it are intensive in close time proximity of the event. However, when the time distance is relatively long, such as a month, there might not be relevant information on Twitter or news regarding the event. The tweet collections were designed with the consideration of time. Each individual collection contains tweets published in a time span of about 10 days, and the queries for testing on the collections are ensured to have some relevant tweets. Therefore, it is reasonable to assume that the events of the queries happened around the time span of the corresponding collections, and it may be appropriate to search the queries only on their corresponding collections. This, in turn, requires the news articles published within or near the same time spans of the tweet collections to be searched on. The Washington Post collection contains news articles published from 2012 to 2017, which overlaps in time with the TREC tweet collections in 2015, 2016, and 2017. Therefore, the tweet collections from these three years are used. Tweet collections used in TREC 2011 and 2012, however, are not included in our experiment [2]. Moreover, given a query from a tweet collection, it is searched only on news articles that are published within the time window from the last day of the time span of the corresponding tweet collection to one month prior to ensure that the news articles aligned with the tweets.

Besides time alignment, a *query filtering* step is also adopted since not all queries can be used. Some of them are not event related. Even for event queries, there might not always be relevant news articles for them, such as the query "Hershey, PA quilt show". Possible explanations are that the news collection is only from one source (Washington Post) which limits the variety of the news being covered. Moreover, some of the events are not significant enough to receive news coverage. Non-event queries can be identified by the event query detection method discussed in the news

[2] TREC 2012 microblog track reused the tweet collection in TREC 2011

retrieval pipeline, whereas the queries without relevant articles can be identified by the *Tree* method mentioned in the silent day detection section, which is proposed to identify queries without relevant information in a news collection. For news retrieval, following the news retrieval pipeline, language modeling is used as the retrieval function. Moreover, the Tree method is trained on *Disk4&5*. After these filtering steps, 18 are left for testing.

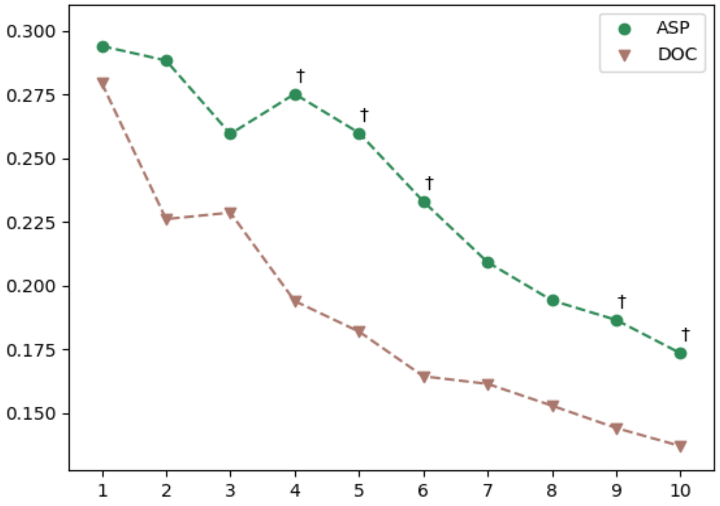
5.2.2.2 Experiment Design

The proposed method involves ranking aspects across news articles and showing the top terms of their language models for users to pick aspects to explore, as well as re-ranking microblog posts for individual aspects. Thus, for it to be useful, several conditions need to be satisfied. First, the aspects should represent potential users' search interests with respect to the event, and their language models should be effective in retrieving microblog posts for these search interests. Additionally, the aspect ranking mechanism should be able to promote the more important aspects that users are likely to be interested in. Last but not least, the top terms of the language models should be able to provide a meaningful understanding of the aspects. To evaluate the last condition, human judges are required to read the top terms of the aspects' language models as well as the tweets to determine the meaningfulness and informativeness of using top terms to represent the aspects. Due to the limit of the resources, we are not able to conduct such an evaluation. Instead, we show examples of top terms of language models and the retrieved relevant tweets for the corresponding aspects to demonstrate that the condition is satisfied. For evaluating the first two conditions, we focus on designing an experiment that uses the relevance judgment as well as semantic clusters of the relevant tweets in the tweet collections.

Semantic clusters are generated by assessors via grouping relevant tweets that cover similar information of the query. These clusters can be viewed as tweet groups representing the same subtopics of the query. If an aspect identified from the news can correlate to a semantic cluster and its language model is effective in assisting the retrieval of the tweets of the cluster, it may certainly prove that the aspect is useful and the estimation of its language model is accurate. Moreover, if this can be observed for top-ranked aspects of an event query, the effectiveness of the aspect ranking algorithm is demonstrated. Thus, the experiment is designed as follows. The methods described in the news retrieval pipeline are used to identify aspects as well as estimate the weights and language models for the aspects. The proposed cross article aspect weighting and aspect based re-ranking methods discussed in this section are used to rank aspects and retrieve tweets for the top N aspects. For each of the N aspect, the retrieval performance measured by ndcg@10 is computed for all semantic clusters of the query, and the one that receives the highest ndcg@10 is picked; the best cluster ndcg@10's are then averaged among the top N aspects to serve as a measure of the overall effectiveness of the proposed aspect based microblog re-ranking paradigm. The measure is called *cluster average ndcg@10*. It is clear that if the measure is high, it may indicate that the aspects correctly correspond to the semantic clusters, and the language models are useful in retrieving tweets for the aspects.

To conduct this experiment, following the previous retrieval settings on the tweet collections, f2exp is used. Top 100 tweets retrieved by f2exp for the queries are used for per-aspect re-ranking. Moreover, the parameters of the proposed paradigm, such as the weight regulator γ in Equation 5.8 that controls the influence of the original query and the aspect language model, as well as λ for estimating language models in L-PLSA, are set using 5-fold cross-validation. The number of top aspects N varies from 1 to 10 with a step size of 1 to cover the likely range of the number of aspects that a user might be interested in. Due to the fact that our method relies on aspects from the news, it is called *ASP* for brevity. In addition, a baseline is implemented to compare the proposed method against. The baseline is similar to the proposed method except the fact that it uses top N *documents* instead of aspects for re-ranking tweets, and the weights of a document is decided by the retrieval score of the document on the news collection (same as in Equation 5.9 without the aspect weight $Pr\left(a_i \mid Q, D\right)$). Since documents are used, this baseline is called *DOC*.

Figure 5.8: Cluster average ndcg@10 for different number of top aspects N



5.2.3 Evaluation

The cluster average ndcg@10 for top N aspects/documents are plotted in Figure 5.8 in which the † symbol indicates that the difference between the two methods at an N value is statistically significant according to the Wilcoxon signed-rank at the level of 0.05. As can be seen, ASP outperforms DOC for every N value. Moreover, the improvements are statistically significant for half of the N values. This may certainly suggest that not only can aspects correspond to subtopics of the event queries, but also that the language models of the aspects are effective in re-ranking the tweets for the aspects. In addition, the generally downward trend of the cluster average ndcg@10 may suggest that the cross-article aspect weighting method is able to assign higher weights to the more important aspects.

Besides the quantitative evaluation conducted above, examples of top terms of the language models of aspects and the top tweets re-ranked for the aspects that are also relevant are shown to illustrate the usefulness of the top terms for helping the users to understand the aspects. More specifically, we show examples of queries, the top 5 terms of the aspect language models of the queries, and the relevant tweets among the top results re-ranked for the aspects in Table 5.6. The common words between the top 5 terms and relevant tweets are highlighted. As can be seen, the top terms are indeed informative. For the first query “Philippines Marawi ISIS”, which is about the battle between the Philippine and ISIS in the city of Marawi, which is known as the Siege of Marawi [3]. The top terms of the two example aspects indicate that the aspects are about the involvement of the Philippines President Duterte in the war and the details of the siege, respectively. This also reflects on the retrieved relevant tweets of the aspects. Similarly, for the query “drones vs. commercial airliners”, the top terms of the example aspect shows that it is about the Federal Aviation Administration regulation of the drones. The example relevant tweet, as a result, mentions newly proposed federal laws about drones.

5.3 Summary

In this chapter, our two major efforts for improving microblog retrieval for event related queries are discussed. First, we propose query collectivity measure, a relevance signal focusing on the collective presence of the query terms, and its two realizations phrase based weighted information gain and local query coherence. The measure is then used to improve the general microblog retrieval, including silent day detection and pseudo relevance feedback. Silent day detection is formulated as a classification problem and the proposed signals are used as features. Experiment results show the effectiveness of them as they outperform the feature sets of existing query performance predictors. Moreover, they also exhibit superiority against the state-of-the-art missing content detection method. In terms of pseudo relevance feedback, on the other hand, the proposed signals are used to select expansion terms in a more conservative fashion. We experimentally illustrate that the proposed signals are able to select terms that are truly related to query terms as the variant LQC_b significantly improves over the non-feedback baseline as well as a state-of-the-art baseline RM3 on queries when the signal could find expansion terms. Two ways of combining LQC_b with RM3 to perform are

[3] https://en.wikipedia.org/wiki/Battle_of_Marawi

Table 5.6: Examples of top terms of aspect language models and the top-ranked tweets for the aspects

Query	Top 5 Terms	Example Relevant Tweets
Philippines Marawi ISIS	**Duterte** Philippines president **Marawi** **war**	groupies, fist-bump greet **Duterte** in visit to **war** zone in **Marawi** **war**
		Duterte to troops on second visit to **Marawi**: 'stay alive, fight cool'
	Philippine **Marawi** **militants** besiege martial	**Philippines**: only 60 **militants** fighting in **Marawi** siege
drones vs. commercial airliners	**drones** **FAA** US aviation airspace	**US** commercial #**drones** face complex #regulation under **FAA** reg's, new proposed fed law http://t.co/kjnsykyx3b

also introduced to further improve the retrieval effectiveness based on either the length of the query or whether LQC_b can find any expansion terms. Both of them are shown to be useful and their improvements over the non-feedback baseline are statistically significant.

Besides improving microblog retrieval in general, we also designed a method to leverage the background aspects from news articles to bridge the news retrieval and microblog retrieval for event queries specifically. To accomplish that, a ranking function is introduced to incorporate both the keyword queries of events and aspects from news to re-rank microblogs for each aspect. Moreover, an aspect weighting scheme is proposed to compare and rank the aspects from different articles so that more important aspects can be prioritized. Experiments are designed to combine an existing news collection with tweet collections with several processing steps such as time alignment and query filtering. Semantic clusters of the tweet collections are used to evaluate the proposed cross-genre search paradigm and the results seem to indicate the usefulness of the aspect based re-ranking method and the aspect ranking scheme. In addition, examples of the top terms of the language models of the aspects as well as the relevant tweets of the semantic cluster corresponding to the aspects are shown, which seem to suggest that presenting the top terms can further assist users in choosing aspects to explore by helping users to understand the aspects.

Chapter 6

CONCLUSION AND FUTURE WORK

In this thesis, we propose a unified framework to support event related information seeking on news and microblogs. It is built on top of the existing ad-hoc retrieval paradigm but includes novel techniques to address the different challenges for event related retrieval on these two different genres.

In event related news retrieval, the importance of background information is discussed. Two methods are proposed to effectively retrieve background information, which are a simple time filter and the aspect based background retrieval model. The time filter is motivated by our intuition that background information is likely to be information in the "past". Statistical analysis validates our intuition. Moreover, experiment results show that the time filter is indeed helpful.

In addition, we also propose to use the background aspects of the event articles to retrieve background information. More specifically, queries are searched on news and event articles are detected by a simple rule based method that uses the titles of articles. Aspects in the event articles are identified by an entity graph based method and the method also produces paragraph labels of the aspects. These labels are then used to estimate the language models of the aspects. Besides, in order to estimate the importance of the aspects, we propose two interpretations of the aspect importance, which are clarity and relatedness. Experiments are conducted on two years' of TREC News Track data. The results show that the rule based event article identification method can accurately identify event articles with high recall. Moreover, various label assisted language model estimation methods, as well as different implementations of the aspect weighting methods, are tested, which illustrates that with appropriate selection of the implementations of these two methods, statistically significant improvement can be achieved for background retrieval on event articles over a competitive baseline. When the aspect based background retrieval model is combined with the time filter, further improvements are observed. This suggests that not only are these two methods effective in retrieving background information, but their improvements are also additive.

Besides the news retrieval pipeline, the microblog retrieval pipeline is the other important part of the proposed framework. In this pipeline, we propose a novel relevance signal called query collectivity measure to improve microblog retrieval in general. Compared with classic relevance signals, the query collectivity measure is a stricter relevance signal that uses the collective presence of multiple query terms instead of the appearance of individual terms to infer relevance. This characteristic enables it to be more capable of distinguishing relevant information from noise.

Two instantiations of it, which are phrase based weighted information gain and local query coherence are proposed. They are applied to silent day detection as well as pseudo relevance feedback for microblog retrieval. Promising results are shown on both tasks. For silent day detection, the proposed measures are used as features and they can outperform the feature set consisting of state-of-the-art query performance prediction methods. Moreover, they also seem to be advantageous against the state-of-the-art method for missing content detection, which is a similar task on document collections.

For pseudo relevance feedback, on the other hand, the proposed measures are used to directly select expansion terms from the initial retrieved results. Experiments show that it is a conservative method that only select expansion terms for a subset of queries. For this subset, performance improvement is observed over a non-feedback baseline and a state-of-the-art pseudo relevance feedback baseline RM3. Further experiments indicate that the proposed method can be combined with RM3 to choose the appropriate feedback method for different queries. Such combinations lead to statistically significant improvement in average for all queries.

Not only do we try to enhance the news and microblog retrieval individually, we also show that aspects from the news can be useful in retrieving microblog posts for the background aspects. More specifically, language models of the aspects are used to re-rank the initial microblog retrieval results for the aspects. We also propose a cross-article aspect weighting method to ensure the weights among aspects from different articles are comparable. These weights are then used to rank the aspects to help users to pick more important aspects. To further assists the selection of aspects, we also hypothesize that showing the top terms of the language model would be helpful since it could provide some basic level of understanding of the aspects. Experiments are conducted on the combination of existing news and microblog TREC collections with extra processing steps. Results seem to suggest that the news aspects can indeed be used to re-rank and organize the microblog posts in a meaningful way. Examples of top terms of aspect language models as well as relevant tweets suggest that the top terms can provide a sensible understanding of the aspects to the users.

There can be multiple interesting research directions for the two types of search for event related queries. On news retrieval, for instance, how to retrieve background information for an aspect that is missing in the event article can be complementary to this work since we only investigate how to leverage the existing aspects. Because, according to the definition of background information, it could be connections of the reported event to a related event, it might be interesting to perform a fuzzy search based on the "five w's" of an event to discover the new aspects. For instance, given an event, another similar event that happens at the same location but at a different time, or involves the same action conducted by the same people but at a different location, might be related and therefore used as an aspect.

Moreover, the time filter is applied to ensure the retrieved background articles are published before that of the event article. It would be interesting to analyze the differences between the background articles published before and after the event articles, and whether the differences can lead to different solutions.

On the microblog retrieval side, although we indirectly prove the usefulness of news aspects in providing the option of finer exploration for microblog data, it would be interesting to conduct user studies to directly evaluate the proposed method, such as how helpful the aspect based re-ranking is in helping users navigate through the microblog data. Moreover, it also would be interesting to investigate how to create aspects besides the background aspects from news to cover a broader range of potential user interests. For instance, sentiment analysis and clustering might be used to create opinion aspects, which can be complementary to the background aspects that are fact-focused.

BIBLIOGRAPHY

[1] Nasreen Abdul-jaleel, James Allan, Bruce W. Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. Umass at trec 2004: Novelty and hard. In *Proceedings of TREC-13*, 2004.

[2] Avi Arampatzis and Jaap Kamps. An empirical study of query specificity. In *Proceedings of ECIR '10*, 2010.

[3] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009.

[4] Lucas Bechberger, Maria Schmidt, Alex Waibel, and Marcello Federico. Personalized news event retrieval for small talk in social dialog systems. In *Speech Communication; 12. ITG Symposium*, 2016.

[5] Agra Bimantara, Michelle Blau, Kevin Engelhardt, Johannes Gerwert, Tobias Gottschalk, Philipp Lukosz, Shenna Piri, Nima Saken Shaft, and Klaus Berberich. htw saar trec 2018 news track. In *Proceedings of TREC 2018*, 2018.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] Christopher Boston, Hui Fang, Sandra Carberry, Hao Wu, and Xitong Liu. Wikimantic: Toward effective disambiguation and expansion of queries. *Data Knowl. Eng.*, 90:22–37, 2014.

[8] Florian Boudin. A comparison of centrality measures for graph-based keyphrase extraction. In *Proceedings of the sixth international joint conference on natural language processing*, 2013.

[9] Jack G. Conrad and Michael Bender. Semi-supervised events clustering in news retrieval. In *Proceedings of the NewsIR'16 Workshop at ECIR*, 2016.

[10] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of SIGIR '02*, 2002.

[11] Ronan Cummins. Document score distribution models for query performance inference and prediction. *ACM Trans. Inf. Syst.*, 32(1), January 2014.

[12] Ronan Cummins, Joemon Jose, and Colm O'Riordan. Improved query performance prediction using standard deviation. In *Proceedings of SIGIR '11*, 2011.

[13] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] Yuyang Ding, Xiaoying Lian, Houquan Zhou, Zhaoge Liu, Hanxing Ding, and Zhongni Hou. Ictnet at trec 2019 news track. In *Proceedings of TREC '19*, 2019.

[16] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of SIGIR '05*, 2005.

[17] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of SIGIR '06*, 2006.

[18] Walter Fox. *Writing the news : a guide for print journalists*, chapter 7, pages 100–106. Iowa State University Press, 3 edition, 2001.

[19] Walter Fox. *Writing the news : a guide for print journalists*, chapter 7, pages 43–44. Iowa State University Press, 3 edition, 2001.

[20] Claudia Hauff, Leif Azzopardi, and Djoerd Hiemstra. The combination and evaluation of query performance prediction methods. In *Proceedings of ECIR '09*, 2009.

[21] Thomas Hofmann. Probabilistic latent semantic analysis. *In Proceedings of SIGIR '99*, 1999.

[22] Nirmal Jonnalagedda and Susan Gauch. Personalized news recommendation using twitter. 3, 2013.

[23] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 1999.

[24] Jimmy Lin, Salman Mohammed, Royal Sequiera, Luchen Tan, Nimesh Ghelani, and Mustafa Abualsaud. Overview of the trec 2017 real-time summarization track. In *Proceedings of TREC '17*, 2017.

[25] Jimmy J. Lin, Miles Efron, Yulu Wang, Garrick Sherman, and Ellen Voorhees. Overview of the trec-2015 microblog track. In *Proceedings of TREC '15*, 2015.

[26] Jimmy J. Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. Overview of the trec 2016 real-time summarization track. In *Proceedings of TREC '16*, 2016.

[27] Marina Litvak and Mark Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization*, 2008.

[28] Michael McCandless, Erik Hatcher, Otis Gospodnetić, and O Gospodnetić. *Lucene in action*, volume 2. Manning Greenwich, 2010.

[29] Qiaozhu Mei and ChengXiang Zhai. A note on em algorithm for probabilistic latent semantic analysis. In *Proceedings of the International Conference on Information and Knowledge Management, CIKM*, 2001.

[30] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005.

[31] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[32] Sondess Missaoui, Andrew MacFarlane, Stephann Makri, and Marisela Gutierrez-Lopez. Dminr at trec news track. In *Proceedings of TREC '19*, 2019.

[33] Taiki Miyanishi, Kazuhiro Seki, and Kuniaki Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, 2013.

[34] Bilel Moulahi, Lamjed Ben Jabeur, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. Irit at trec real-time summarization 2016. In *Proceedings of TREC '16*, 2016.

[35] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier information retrieval platform. In *Advances in Information Retrieval*, 2005.

[36] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the trec-2011 microblog track. In *Proceedings of TREC '11*, 2011.

[37] Larry Page, Sergey Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[38] Joaquín Pérez-Iglesias and Lourdes Araujo. Standard deviation as a query hardness estimator. In *Proceedings of SPIRE '10*, 2010.

[39] "#blacklivesmatter surges on twitter after george floyd's death". Pew Research Center, Washington, D.C., Jun. 10 2020. https://www.pewresearch.org/fact-tank/2020/06/10/blacklivesmatter-surges-on-twitter-after-george-floyds-death/.

[40] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, 2009.

[41] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009.

[42] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009.

[43] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In *NIST Special Publication 500-225: Overview of the Third Text REtrieval Conference (TREC-3)*, 1994.

[44] Jesus A. Rodriguez Perez and Joemon M. Jose. Predicting query performance in microblog retrieval. In *Proceedings of SIGIR '14*, 2014.

[45] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[46] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. Predicting query performance by query-drift estimation. *ACM Trans. Inf. Syst.*, 30(2):11:1–11:35, May 2012.

[47] Ian Soboroff, Shudong Huang, and Donna Harman. Trec 2018 news track overview. In *Proceedings of TREC '18*, 2018.

[48] Ian Soboroff, Iadh Ounis, Craig Macdonald, and Jimmy Lin. Overview of the trec 2012 microblog track. In *Proceedings of TREC '12*, 2012.

[49] Fei Song and W. Bruce Croft. A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, 1999.

[50] Ashok Srivastava and Mehran Sahami. *Text Mining: Classification, Clustering, and Applications*. CRC Press, 2009.

[51] Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, 2005.

[52] Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles L.A. Clarke. An exploration of evaluation metrics for mobile push notifications. In *Proceedings of SIGIR '16*, 2016.

[53] Jinzhi Tang, Chao Lv, Lili Yao, and Dongyan Zhao. PKUICST at TREC 2017 real-time summarization track: Push notifications and email digest. In *Proceedings of TREC '17*, 2017.

[54] Stephen Tomlinson. Robust, web and terabyte retrieval with hummingbird search-server tm at trec 2004. In *Proceedings of TREC-13*, 2004.

[55] Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Linking online news and social media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 2011.

[56] Yue Wang, Hao Wu, and Hui Fang. An exploration of tie-breaking for microblog retrieval. In *Advances in Information Retrieval*, 2014.

[57] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[58] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically examining the "neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[59] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proceedings of SIGIR '05*, 2005.

[60] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2), April 2004.

[61] Ying Zhao, Falk Scholer, and Yohannes Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proceedings of ECIR '08*, 2008.

[62] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *Proceedings of SIGIR '07*, 2007.

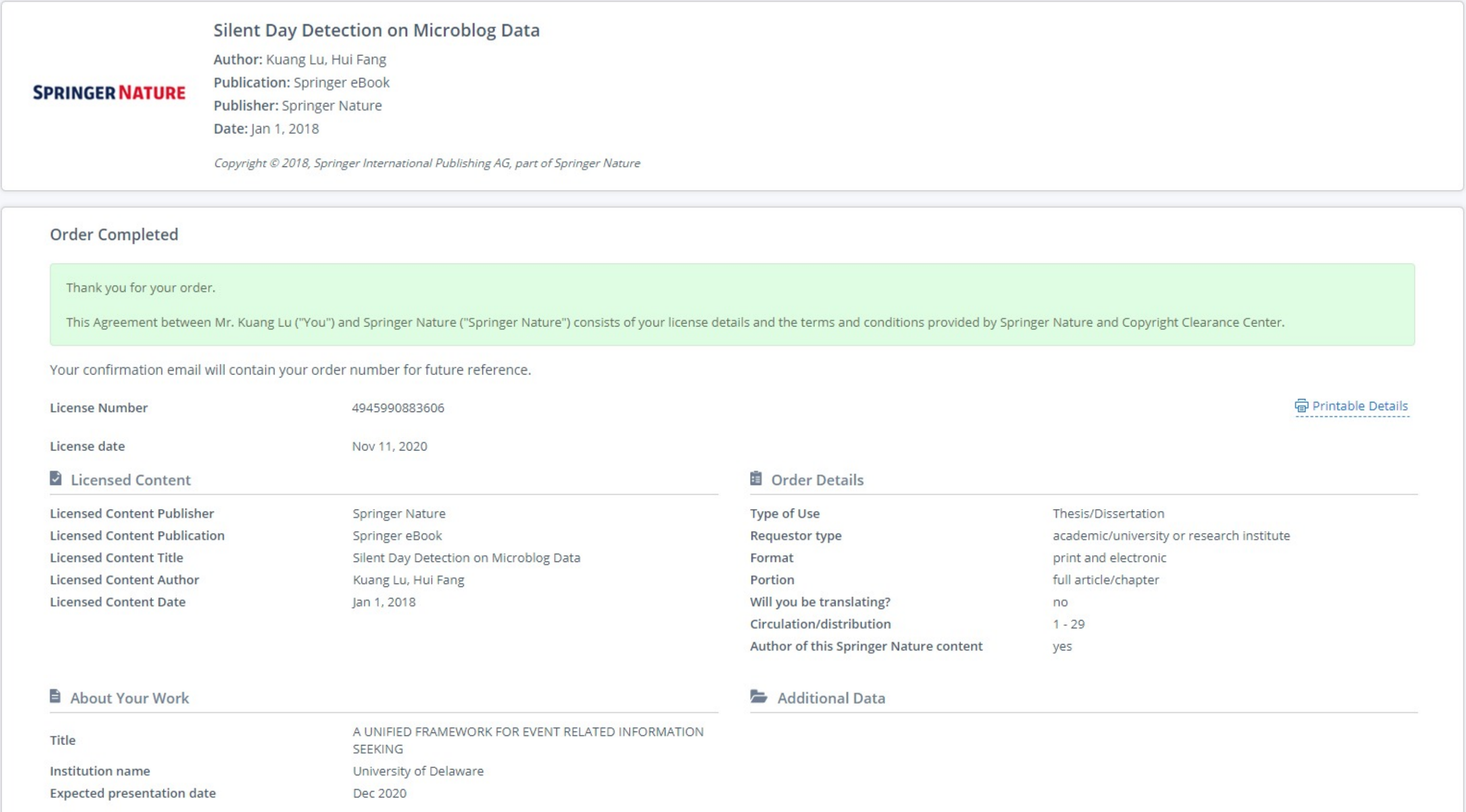
[63] Xiang Zhu, Jiuming Huang, Sheng Zhu, Ming Chen, Chenlu Zhang, Zhenzhen Li, Huang Dongchuan, Zhao Chengliang, Aiping Li, and Yan Jia. NUDTSNA at TREC 2015 microblog track: A live retrieval system framework for social network based on semantic expansion and quality model. In *Proceedings of TREC '15*, 2015.

[64] Xiang Zhu, Jiuming Huang, Sheng Zhu, Ming Chen, Chenlu Zhang, Li Zhenzhen, Huang Dongchuan, Zhao Chengliang, Aiping Li, and Yan Jia. NUDTSNA at TREC 2015 microblog track: A live retrieval system framework for social network based on semantic expansion and quality model. In *Proceedings of TREC '15*, 2015.

[65] Meriem Amina Zingla, Latiri Chiraz, and Yahya Slimani. Short query expansion for microblog retrieval. *Procedia Comput. Sci.*, 96(C):225–234, October 2016.

Appendix

COPYRIGHT PERMISSIONS



SPRINGER NATURE

Silent Day Detection on Microblog Data

Author: Kuang Lu, Hui Fang
Publication: Springer eBook
Publisher: Springer Nature
Date: Jan 1, 2018

Copyright © 2018, Springer International Publishing AG, part of Springer Nature

Order Completed

Thank you for your order.

This Agreement between Mr. Kuang Lu ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

License Number	4945990883606
License date	Nov 11, 2020

Licensed Content

Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Silent Day Detection on Microblog Data
Licensed Content Author	Kuang Lu, Hui Fang
Licensed Content Date	Jan 1, 2018

Order Details

Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes

About Your Work

Title	A UNIFIED FRAMEWORK FOR EVENT RELATED INFORMATION SEEKING
Institution name	University of Delaware
Expected presentation date	Dec 2020

Additional Data