Event Oriented Query Expansion for News Event Queries

Kuang Lu and Hui Fang

Department of Electrical and Computer Engineering
University of Delaware
140 Evans Hall, Newark, Delaware, 19716, USA
{lukuang,hfang}@udel.edu

Abstract. Query expansion techniques have been commonly used by participants in Temporal Summarization tack. However, many previous attempts focused on expanding queries based on their event types, which only covers partially of the information need represented by queries. The reason is that the queries in the TS track are about news events, and for an event query, the event related entities, which are the entities mentioned in the queries, are essential when defining the event. Given the query "Vauxhall helicopter crash", without "Vauxhall" or "helicopter", the event cannot be specifically defined. Therefore, we argue that both event type and event related entities should be considered when expanding a query, and a model based query expansion framework is proposed which incorporates these two types of information. The framework is then employed by using external resources, such as external corpora and Wikipedia pages to build expansion models.

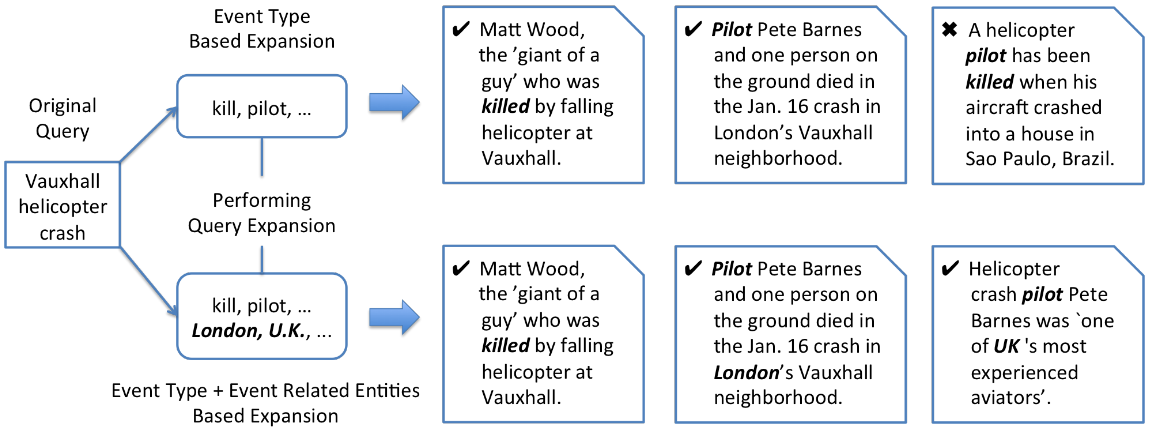
1 Introduction

Temporal Summarization track aims to develop systems for efficiently monitoring the information associated with an event over time, and the queries provided are about news events, such as protests, accidents or natural disasters[1]. Since the start of the track, query expansion is employed by many participants[2] [3] and seems to be essential. However, previous participants tend to use only event type related information to expand queries, meaning that they tried to find information usually used to describe an event type to obtain a richer query representation. Although this type of approach is indeed helpful since it reduces the vocabulary gap between the query and relevant documents/sentences, it is still insufficient for the event queries since it lacks of the information about another important aspect: event related entities. Here, event related entities are the entities mentioned in the query. They contain location, parties involved in the event and other information to a query that is essential for identifying the event of the query. Intuitively, when describing an event, usually two things are mentioned: event type and event related entities. Event type offers general information about what the event is, whereas the event related entities can further specify the event instance. Therefore, these entities should also be considered when expanding event queries. For instance, for the query "Vauxhall helicopter crash", terms very related to the entity "Vauxhall"(e.g. London) can be very useful for capturing relevant information, since they are also likely to occur in relevant documents and sentences. Furthermore, information about event related entities could also help systems distinguishing between documents/sentences describing different events with the same event type. For instance, given the same query as above, if only event type information(e.g. air transport accident related terms such as "kill") is used to expand the query, a system could hardly distinguish truly relevant sentences from sentences describing other concurrently happening air transport accidents. Therefore, it can be argued that expanding queries with information about event-related entities in the query can be used to capture more aspects of queries and thus improve the performance.

In this year, we explore the potential of query expansion for event queries. Specifically, we identify two types of expansion information resources: event type information and event related entities. A language model based query expansion framework is built which incorporates these two types of information. Our system for the Temporal Summarization track employs this framework and uses auxiliary corpora and Wikipedia as the real sources of the two types of expansion information.

2 Query Expansion Framework for Event Queries

In order to fully expand an event query with all its aspects, we identify two types of information that should be used for query expansion: event type and event related entities. For event type, we could find terms usually used for describing the type of event that a query belongs to, and use the terms to expand the query. For instance, given a query "Vauxhall helicopter crash", if terms describing air transport accident, such as "kill" and "pilot", are added to the query, it is likely that a system could find relevant information more effectively and accurately. For event related entities, which are the entities mentioned in queries, the terms used to describe these entities are likely to appear in relevant documents/sentences. For example, given the same query as above, there were other traffic accidents happening concurrently with this helicopter crash. Adding terms related to event related entities of this event instance, such as "London" and "UK", to the query model could help our system better distinguishes sentences related to the query from sentences related to other accident instances. This effect can be illustrated in Figure 2.



✔ Matt Wood, the 'giant of a guy' who was ***killed*** by falling helicopter at Vauxhall.

✔ ***Pilot*** Pete Barnes and one person on the ground died in the Jan. 16 crash in London's Vauxhall neighborhood.

✖ A helicopter ***pilot*** has been ***killed*** when his aircraft crashed into a house in Sao Paulo, Brazil.

✔ Matt Wood, the 'giant of a guy' who was ***killed*** by falling helicopter at Vauxhall.

✔ ***Pilot*** Pete Barnes and one person on the ground died in the Jan. 16 crash in ***London***'s Vauxhall neighborhood.

✔ Helicopter crash ***pilot*** Pete Barnes was `one of ***UK*** 's most experienced aviators'.

Fig. 1. Adding Information about Event Related Entity Helps Improving System Performance

Model-based feedback has been shown effective by previous studies[4] [5]. Based on these studies, we propose a query expansion framework such that the expansion models come from both event type and event related entities. Our framework can be illustrated as Equation 1. For some query Q with event type T and a set of entities E, $P(w \mid \hat{\theta}_Q)$ denotes the estimate of the true query model. $P(w \mid \tilde{\theta}_Q)$ is the maximum likelihood estimation of the query model using original query string. $P(w \mid \theta_T)$ is the estimate of the language model of event type T. $P(w \mid \theta_E)$ is the mixture of estimate language models of the event related entities. Note that $\alpha + \beta + \gamma = 1$. If γ is set to zero, the expansion model is only about event type. Whereas if β is set to zero, only event related entities contributes to the expansion model.

The model of event related entities can be estimated using Equation 2. e_i is an event related entity mentioned in the query and $P(w \mid e_i)$ is the language model usually used to describe the entity, $|E|$ is the size of E, meaning the number of entities in the query.

$$P(w \mid \hat{\theta}_Q) = \alpha P(w \mid \tilde{\theta}_Q) + \beta P(w \mid \theta_T) + \gamma P(w \mid \theta_E) \tag{1}$$

$$P(w \mid \theta_E) = \sum_{e_i \in E} \frac{1}{|E|} P(w \mid e_i) \tag{2}$$

3 Method Description

Although we submit multiple runs for task two and three of this year's TS track, the general framework of our methods is similar as shown in Figure 3. First, we pre-process the corpus to extract information such as sentence text, sentence id, document id, as well as document time, and the documents are stored in hourly batches for further process; second, we employ the query expansion model described above and estimate expansion model from external resources to build rich query representation and use them to compute relevance scores of documents and sentences; finally, relevant documents are selected, and from them, relevant sentences are selected with redundant sentences removed. Note that document selection is not always performed since for task three, all documents in the sub-corpus are relevant.

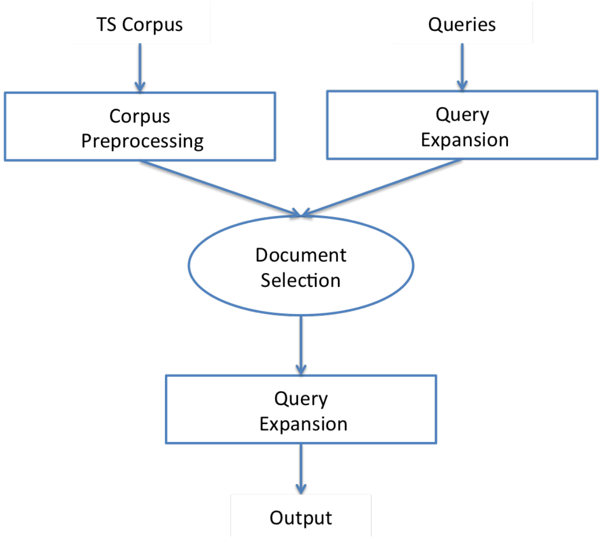


Fig. 2. General System Framework

3.1 Corpus Pre-processing

Since the corpus is stored in the Thrift format and contains information that might not be useful to the task, such as original html content, we chose to preprocess the corpus to store only the information that is used by us. First, we filtered out documents that are not news articles, as done by some participants of previous TS tracks[2], since the sentences in news articles are well written and we do not need to worry about the documents being spam web pages. After that, for every document, only its document id, timestamp, sentence text and sentence id are extracted for latter usage. Porter stemming is performed on all sentences before they are stored. Because we decide to generate updates hourly, documents with timestamps of the same hour are stored into the same file to ease further process.

3.2 Query Expansion Implementation

In order to avoid using future information, we decided to use external resources to build expansion models. Using external resources is shown to be effective for query expansion by previous study[5][6]. We use two types of external resources: external corpora and Wikipedia pages. For the external corpora, some of them are previous trec corpora: the AQUAINT corpus[1] which is divided into three corpora (apw, nyt,

[1] http://www-nlpir.nist.gov/projects/aquaint/

xie) according to the news agencies, trec-ap corpus[2] and Wall Street Journal corpus[3]. We also used a corpus of New York Times pages crawled from 2006 to 2011. These corpora consist of only news articles and thus the quality of the document text is high. Moreover, for the Wikipedia pages, we automatically crawled some Wikipedia page revisions according to the original queries. What are the pages that we crawled will be discussed later. It is important to note that we only use the documents and Wikipedia page revisions strictly before the query start time so that no future information is used.

In order to find event type related information, we first need to define event types for queries. Since the queries in the Temporal Summarization track are mostly disasters, we use and the disaster classification from an online disaster database maintained by the Center for Research on the Epidemiology of Disasters (CRED) [7] as well as the *query* and *type* field of the queries to manually define finer event types for all queries, as shown in Table 1. To expand queries using event type related information, we employ a method similar to [5]. Specifically, for each event type, its name is used as query to search against all corpora. For every corpus, top ranked documents are used to estimate a language model. We take the average of these relevance models to form an expansion model for the event type with the term weights discounted by the corpus frequency of the term, meaning that the less the number of corpora a term occurs, the more its weigh is punished. Basically, we want to degrade the effect of the terms which are only frequent in documents in some corpora but not in others. For example, the term “China” in the Xinhua corpus has very high weight, which is probably caused by the fact that Xinhua is a news agency in China. However, adding the term to queries might not help improving performance. Being able to punish such terms is also the reason why we use multiple corpora together. For each of this language model related to a certain event type, we call it the ”profile” of the event type. Each query will have a event type profile associated with it, and the profile is added to the final query model for relevance calculation.

For estimating language models of event related entities, we use Wikipedia as the source. Specifically, inspired by[6], we first use exact match to find the sub query strings that are the titles of some Wikipedia pages, and use them as event related entities. For every one of such Wikipedia pages, we find all other Wikipedia pages referred to, or referred by the page and crawled the content of the revisions of these pages as well as the original page. The original query substring is then used as the query to rank these pages, and top ranked pages were used to build a language model about the substring. All the language models of the querys substrings are combined to expand the query. The rational behind this method is that the top ranked Wikipedia pages for an event related entity contain considerable text that related to the entity, and thus it is reasonable to estimate the language of the entity from these Wikipedia articles. Moreover, it is also important to note that since some of the query entities can also be related to the disaster event, such as “explosion” in query “brazzaville explosion”, and therefore the query expansion model generated by them are actually about event types. Such expansion models are also helpful in finding relevant information, and thus we did not try to distinguish these entities from others. We call the expansion model we obtain from Wikipedia pages as “Wikipedia expansion models”. Note that, in order to avoid using future information, we used a dump of the Wikipedia page link information that was generate by dbpedia on March, 2010[4] to detect query entities and find related pages. Moreover, all the crawled revisions of the Wikipedia pages were written before the query start time.

3.3 Document and Sentence Selection

The way we select relevant documents and sentences are very similar. Since we used language modeling based query expansion and documents and sentences are both represented as language models, it is reasonable to use the same method, which is query likelihood with Dirichlet smoothing [8], to rank them. One of the differences is that the document relevance scores are only computed by using documents, but the relevance scores of sentences are computed by using both sentences and the documents containing them. In other words, the sentence relevance score is a linear interpolation between relevance scores

[2] http://www.daviddlewis.com/resources/testcollections/trecap/
[3] https://catalog.ldc.upenn.edu/LDC93T3A
[4] http://dbpedia.org/data-set-36

Table 1. The Event Types of Queries

Query	Event Type
26	air transport accident
27	tropical storms
28, 38	collapse
29, 34, 35, 36 ,37,39, 40, 41	bombing
30	explosion
31	power explosion
32, 46	protest
33	conflict
42	fire
43	water transport accident
44, 45	earthquake

computed against document model and sentence model. It is reasonable since we argue that sentences in a relevance document are more likely to be relevant.

Another difference between document and sentence selection is the selection cutoff. For document selection, we chose to use top ten documents. Experiments on last years data showed that for various expansion models, using top 10 documents can cover more than 90% of nuggets. We think such nugget recall is sufficient for generating sentence updates with reasonable performance. Moreover, the number of documents is also small enough to speed up sentence selection. However, the relevance score of sentences are very sensitive to what types expansion models are used, and it is not reasonable to set a fixed number as the cutoff since the number of relevant sentences is largely various among different queries. Therefore, for different runs we tuned the parameters of retrieval model as well as the relevance score threshold on last years data and apply them to this year 's methods. After finding candidate sentences, redundant sentences need to be removed. We use simple cosine similarity and also tuned the similarity threshold on last year 's data.

4 Submissions

In this years TS track, there are three tasks and the difference is only the corpus. We chose to only participate in the second and third task since the sizes of the corpora are much smaller. All the parameters of the runs are tuned in previous year's data. The detailed description for our submissions are explained below:

4.1 Task 2

For task two which has a pre-filtered corpus with smaller number of documents that are more likely to be relevant, we submitted three runs. Since the aim of us in this year's track is to examine how effective our expansion framework is, the only difference between the three runs is the expansion model:

- **WikiProfMixFS:** In this run, we use the both event type profile and Wikipedia expansion model to expand the queries.
- **WikiOnlyFS:** We only use Wikipedia expansion model in this run.
- **ProfOnlyFS:** Only event type profile is used.

4.2 Task 3

For task three, the documents in the sub corpus is ensured to be relevant. We also choose to submit three runs. They are similar to the runs we submitted to task two without performing document filtering, since that is not necessary:

- **WikiProfMix:** In this run, we use the both event type profile and Wikipedia expansion model to expand the queries.
- **WikiOnly:** We only use Wikipedia expansion model in this run.
- **ProfOnly:** Only event type profile is used.

5 Results and Analysis

6 Conclusion

References

1. Aslam, J., Diaz, F., Ekstrand-Abueg, M., McCreadie, R., Pavlu, V., Sakai, T.: Trec 2014 temporal summarization track overview. In: Proceedings of the 2014 TREC conference. (2014)
2. Liu, Q., Liu, Y., Wu, D., Cheng, X.: Ictnet at temporal summarization track trec 2013. In: Proceedings of the 2013 TREC conference. (2013)
3. Xu, T., McNamee, P., Oard, W.D.: Hltcoe at trec 2013: Temporal summarization. In: Proceedings of the 2013 TREC conference. (2013)
4. Zhai, C., Lafferty, J.: Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In: CIKM. (2001)
5. Diaz, F., Metzler, D.: Improving the estimation of relevance models using large external corpora. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '06, New York, NY, USA, ACM (2006) 154–161
6. Aggarwal, N., Buitelaar, P.: Query expansion using wikipedia and dbpedia. In Forner, P., Karlgren, J., Womser-Hacker, C., eds.: CLEF (Online Working Notes/Labs/Workshop). (2012)
7. D. Guha-Sapir, R. Below, P.H.: Em-dat: International disaster database www.emdat.be Universit Catholique de Louvain Brussels Belgium.
8. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22**(2) (April 2004) 179–214