BIT and Purdue at TREC-KBA-CCR Track 2014*

Jingang Wang[1], Ning Zhang[2], Zhiwei Zhang[2], Dandan Song[1], Luo Si[2], Lejian Liao[1]

{bitwjg, sdd, liaolj}@bit.edu.cn, lemonustc@gmail.com, zwzhang.purdue@gmail.com, lsi@purdue.edu

[1]School of Computer Science, Beijing Institute of Technology, Beijing, China
[2]Department of Computer Science, Purdue University, West Lafayette, USA

Abstract. This report summarizes our participation at KBA-CCR track in TREC 2014. Our submissions are generated in two steps: (1) Filtering a candidate documents collection from the stream corpus for a set of target entities; and (2) Estimating the relevance levels between candidate documents and target entities. Three kinds of approaches are employed in the second step, including query expansion, classification and learning to rank. Query expansion is an unsupervised baseline by combining an entity and its related entities as a query to retrieve its relevant documents. Query expansion performs considerably well in *vital + useful* scenario. It's not difficult to filter a relevant document set from the stream corpus. However, in *vital only* scenario, supervised approaches are more powerful than query expansion in identifying *vital* documents for target entities. Our results reveal that learning to rank approaches are more suitable for CCR with current evaluation methodology.

1 Introduction

Task Description. A CCR system is required to filter a chronological stream corpus to detect relevant documents for a set of Knowledge Base (KB) entities. Unlike traditional information retrieval and filtering tasks, CCR not only retrieve relevant documents from the stream corpus, but also distinguish relevant documents according to their relevance levels to the target entities. The relevance of a document to an entity is represented by a confidence score in the range of (0, 1000]. To evaluate the system performance, a cutoff value is varied from 0 to 1000 and the documents with scores above the cutoff are treated as relevant. Correspondingly, the documents with scores below the cutoff are irrelevant documents. Two measurements are calculated: (i) $max(F(avg(P), avg(R)))$ and (ii) $max(SU)$. SU(Scaled utility) is a metric introduce in [3] to evaluate the ability of a information filtering system to separate relevant and irrelevant documents. Given a cutoff, we could calculate P, R, F and SU respectively for each entity

* This work was done during Jingang's visit at Purdue University.

and obtain the macro-average values of all entities. Given a set of target entities, there are two scenarios: (i) *vital only:* detecting *vital* documents in the stream corpus, and (ii) *vital + useful:* detecting *vital* and *useful* documents in the stream corpus.

Stream Corpus. The stream corpus, a temporally-ordered document collection, contains approximately 20 million documents published during the period from Oct. 2011 to May 2013.

Target Entity. The target entity set is composed of 71 entities found in the stream corpus, including facilities, persons and organizations. Some entities are already contained in Wikipedia, while others do not exist in any existing KB.

Compared with previous tracks, TREC-KBA-CCR 2014 is unique from three aspects.

1. The target entities are selected from the stream corpus itself instead of an existing KB. In previous tracks, all target entities are from either Wikipedia or Twitter.
2. The annotation schema is revised more precisely, including four relevance levels: *vital*, *useful*, *unknown* and *non-referent*. In addition, the annotation quality is improved significantly.
3. The cutoff between training and test is not consistent for different entities. This variation promises each entity exists annotation instances in each relevance levels.

In TREC-KBA-CCR 2014, we submitted 7 runs, including a baseline run, 2 query expansion runs, 2 classification runs and 2 learning to rank (LTR) runs.

The rest of this paper is organized as follows: Section 2 introduces an entity-centric filtering step to reduce the volume of stream corpus. Next, we present our relevance estimation approaches in Section 3. Section 4 introduces the bursty features utilized in our supervised approaches in detail. Finally, we summarize the results of our submissions and come up with some conclusions in Section 5.

2 Filtering

According to the assessment analysis of former KBA tracks, most relevant documents mention the target entity explicitly, and merely 0.4% of documents without referring to the target entity are labeled as relevant [2]. Therefore, it is not indispensable to process all the documents in the corpus, which is extraordinarily time-consuming and laborious. Before further relevance estimation, we undertake an effective entity-centric filtering step to remove obviously irrelevant documents from the stream corpus.

As described in [5], we first index the documents in the stream corpus with *ElasticSearch*, and then filter it with an entity-centric phrase query. To construct a high-recall query to retain as many candidate documents as possible, we need to

expand enough surface forms for each target entity. Freebase[1] hereby is utilized to expand the surface forms for the entities. The entities can not be found in Freebase are not considered as popular entities, so we do not expand it at all.

We formulate a baseline query to filter the stream corpus. Only the matched documents are retained and processed in subsequent process. Given an entity E, the surface form set of E is $Rel(E) = \{E_i | i \in [1, M]\}$, the baseline phrase query for E is

$$E \vee \ E_1 \vee E_2 \vee \cdots \vee E_M, \tag{1}$$

where the $\vee$ operator ensures at least one operand is true, representing the corresponding term is matched in the document.

To evaluate the filtering performance, we calculate the maximum $macro_avg(Recall)$ by setting the cutoff value as 0, in which case all the retrieved documents are considered as positive instances in both scenarios. The results are listed in Table 1. The filtering performance is surprisingly satisfactory, proving that most

Table 1. Recall of baseline expansion

Filtering Method	max(*macro_average*(*R*))	
	Vital	Vital + Useful
baseline query	.987	.985

Note: The recall metrics are calculated with ground truth data (*trec-kba-2014-10-15-ccr-and-ssf.after-cutoff.tsv*) excluding entities without *training_end_date* field.

relevant documents, either *vital* or *useful*, mention the target entity explicitly. The volume of the candidate documents are reduced to less than 1 million after filtering.

3 Relevance Estimation

A candidate document collection is obtained from the stream corpus after filtering, we estimate the relevance between the candidate documents and target entities. We have employed 3 kinds of approaches, including query expansion, classification and LTR.

3.1 Query Expansion

Although the baseline query demonstrated in Equation 1 has achieved considerable filtering performance, it can not estimate the fine-grained relevance level between a document and an entity. Query Expansion (QE) is an effective approach to solve this problem. In our work, we expand the baseline phrase query

[1] https://www.freebase.com/

with contextual entities found in target entities' profiles and annotation data, and then search against the built index to acquire the candidate documents. For example, $\{E_i|i \in [1, N]\ \}$ is the related entity set we found for target entity E, the query is expanded as follows:

$$Baseline \wedge \{E_1 \vee E_2 \vee \cdots \vee E_N\} \tag{2}$$

Baseline represents the baseline query demonstrated in Equation 1. The return documents of the query are relevant documents and the ranking scores are scaled to the final confidence scores.

In addition to the baseline run, we submitted 2 other QE runs: *QE-Profile* and *QE-Labeled*. The differences lies in the query terms they utilized. *QE-Profile* expands the baseline query with the related entities found in entities' profiles, while *QE-Labeled* expands the baseline query with the related entities found in the annotation data.

3.2 Classification

CCR is usually formulated as a binary classification task to distinguish relevant/irrelevant documents (i.e., *vital + useful*) or vital/useful documents (i.e., *vital only*).

We submitted 2 classification runs: *ClassificationV* and *ClassificationU*. The former one classifies the candidate documents into vital or non-vital, yet the latter one classifies them into relevant (*vital + useful*) or irrelevant (*unknown + non-referent*). All the classifiers are implemented with random forest classification model, which was reported as the best classification model in CCR.

Please note that we build a global classifier with all training instances instead of building a local classifier for each entity for simplicity. The classification results would be improved if local classification model could be built for every entity individually.

3.3 Learning to Rank

CCR can be considered as a learning to rank (LTR) task as well because of the intrinsic ordering of different relevance levels, i.e., $vital > useful > unknown > non-refrent$.

We submitted 2 LTR runs: GlobalRank and BinaryRank. First, we build a global ranking model with all training instances. Moreover, we build a local ranking model for the entity with enough training instances . *GlobalRank* rank the test instances with the global ranking model. In terms of *BinaryRank*, if there exists a local model for an entity, we rank the test instances with the local model, otherwise the global model.

4 Features

Classification and LTR are both supervised approaches, for whom we adopt the same semantic feature sets introduced in [5]. CCR is filtering relevant documents from a temporally-ordered stream corpus and entities are evolving with

the passage of time, but the semantic features can not portray the dynamic characteristics of entities in the stream corpus. Temporal features are introduced to make up for this deficiency in previous work [5,1].

We develop bursty features as our temporal features. The underlying intuition is that the appearance of an entity in the stream corpus is signaled by a 'burst of activity', with relevant documents rising sharply in frequency as something important are happening around the target entity. In previous work, Wikipedia daily view statistics are utilized to detect entities' bursty period, during which documents appear is more possible to be relevant than those not. Unfortunately, not all entities are from Wikipedia, we can not adopt the statistics to capture bursty periods. Alternatively, we calculate the bursty periods based on two other statistics. (1) Google Trends[2] is a similar resource we can resort to. It is a public web statistics, based on Google Search, that shows how often a particular search term is entered relative to the total search-volume. Besides, (2) entities' appearances in the stream corpus are also utilized to detect their bursty periods. Figure 1 plots an example of the bursts of Wikipedia entity *Benjamin Bronfman* with two obvious bursty periods over the entire text stream.

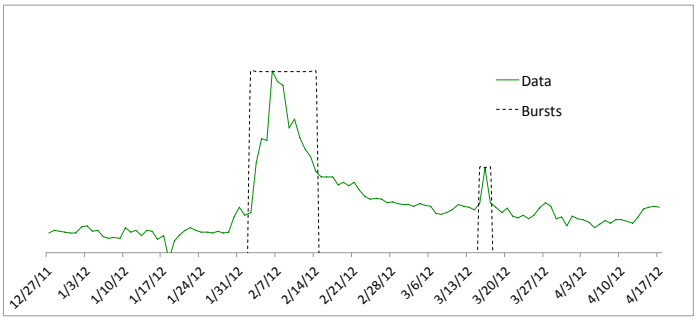

Fig. 1. Example of Bursts of Entity *Benjamin Bronfman*

For entity E, we have a daily view/search statistics sequence $\mathbf{v} = (v_1, v_2, \cdots, v_n)$. We detect E's bursts from $\mathbf{v}$ with a tailored moving average (MA) method [4]. Unlike the moving average method in [4] using a unified cutoff, we calculate an individual cutoff for each moving average (MA). More concretely, for each item v_i in $\mathbf{v}$,

1. Calculate its moving average of length w as $MA_w(i) = \frac{v_i + v_{i-1} + \cdots + v_{i-w+1}}{w}$.
2. Calculate cutoff $c(i)$ based on previous MA sequence $Pre_{MA} = (MA_w(1), \cdots, MA_w(i))$ as
$$c(i) = mean(Pre_{MA}) + \beta \cdot std(Pre_{MA}).$$
3. Bursty day sequence: $\mathbf{d} = \{i | MA_w(i) > c(i)\}$.

[2] http://www.google.com/trends/

4. Calculate daily bursty weights $\mathbf{w} = \{w_i | \frac{MA_w(i)}{c(i)}, i \in \mathbf{d}\}$.

The moving average length can be varied to detect long-term or short-term bursts. We set the moving average length as 7 days (i.e., $w = 7$). The cutoff value is empirically set as 3 times the standard deviation of the MA (i.e., $\beta = 2$). Moreover, we compact the consecutive days in $\mathbf{d}$ into bursty periods. The bursty weight for each period is calculated as the average weight of all the bursts in this period.

4.1 Feature Representation

Given a document D and entity E, we define a bursty value $b(D, E)$ to represent their temporal relations. Let t be the timestamp of D. If t is in E's bursty period $[t_{start}, t_{end}]$, then $b(D, E)$ is calculated as Equation 3 shows. If t is not in any butsy period, $b(D, E)$ would be set as 0.

$$b(D, E) = (1 - \frac{t - t_{start}}{t_{end} - t_{start}}) \cdot bw_{(t_{start}, t_{end})}(E), \quad t \in [t_{start}, t_{end}] \tag{3}$$

$1 - \frac{t - t_{start}}{t_{end} - t_{start}}$ is a decaying coefficient reflecting the intuition that the documents appear at the beginning of a burst are more informative than those appear at the end.

5 Results and Discussion

All the results are listed in Table 2. It's surprising that baseline and Query Expansion are excellent enough to filter the relevant documents from the stream corpus. Nearly all the truth data are detected from the stream corpus. In *vital + useful* scenario, the value of F is dominated by *recall*, and the best performance is achieved when cutoff is equal to 0. This phenomenon also exist in *vital* scenario, all our approaches achieve close values of *macro_average_F*. In *vital only* scenario, LTR methods achieve better *micro_average_F* than classification approaches, and *BinaryRank* achieves the best performance out of all approaches. This reveals that LTR approachs are more suitable for CCR in current evaluation framework. A possible explain is that the ranked output of LTR approaches can be transformed to the desired format of confidence score in a straightforward manner, while the binary output of classification need additional mechanisms to be transformed into the target format.

References

1. Balog, K., Ramampiaro, H.: Cumulative citation recommendation: classification vs. ranking. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 941–944. SIGIR '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2484028.2484151

Table 2. Results of official runs. All the measurements are reported by official scorer with cutoff-step-size=10.

Run	Vital Only				Vital + Useful			
	Macro_Average		Micro_Average		Macro_Average		Micro_Average	
	F	*SU*	*F*	*SU*	*F*	*SU*	*F*	*SU*
baseline	.461	.299	.348	.329	.960	.950	.962	.970
QE-Profile	.461	.321	.360	.336	.960	.950	.962	.970
QE-Labeled	.461	.317	.370	.344	.960	.950	.962	.970
ClassificationU	.464	.301	.344	.295	.951	.938	.955	.963
ClassificationV	.460	.344	.341	.348	.951	.938	.955	.963
BinaryRank	.459	.341	.379	.345	.951	.938	.955	.963
GlobalRank	..460	.343	.373	.348	.951	.938	.955	.963

2. Frank, J.R., Kleiman-Weiner, M., Roberts, D.A., Niu, F., Zhang, C., Re, C., Soboroff, I.: Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. In: Proceedings of the Text REtrieval Conference (TREC) (2012), http://trec.nist.gov/act_part/conference/notebook.papers/KBA.OVERVIEW.pdf
3. Robertson, S., Soboroff, I.: The trec 2002 filtering track report. In: TEXT RETRIEVAL CONFERENCE (2002)
4. Vlachos, M., Meek, C., Vagena, Z., Gunopulos, D.: Identifying similarities, periodicities and bursts for online search queries. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. pp. 131–142. SIGMOD '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1007568.1007586
5. Wang, J., Song, D., Lin, C.Y., Liao, L.: Bit and msra at trec kba ccr track 2013. Notebook of the TExt Retrieval Conference (2013)