A Related Entity based Approach for Knowledge Base Acceleration

Xitong Liu, Jerry Darko, Hui Fang
University of Delaware
Newark, DE, USA
{xtliu,jdarko,hfang}@udel.edu

1 Introduction

In this paper we present the overview of our work in the TREC 2013 KBA Track. The goal is to find documents which may contribute to the update of knowledge base entries (e.g., Wikipedia or Freebase articles). Two tasks are introduced in this year's track: (1) Cumulative Citation Recommendation (CCR), (2) Streaming Slot Filling (SSF). Particularly, we focus on the CCR task, follow our previous work on TREC 2012 KBA Track [3] and propose an improved approach by incorporating weighting to related entities.

Our approach is based on the framework of leveraging related entities for document filtering. In particular, we pool related entities from the profile page (i.e., Wikipedia) of target entity, estimate the weight of each related entities based on the training data, and apply the weighted related entities to estimate the confidence scores of streaming documents. We explore three methods based on different weighting strategies: (1) all related entities get zero weight, which is equivalent to exact matching against the target entity only. (2) all related entities get same weight. (3) related entities are weighted based on training data. Experiments on the KBA Stream Corpus 2013 reveal the effectiveness of our approach.

2 Data Preprocessing

The CCR task aims to find relevant documents worth citing for the profile of a given entity (e.g., Wikipedia or Freebase entry). In this year 170 target entities are provided, among which 150 are from Wikipedia (specified by URL of Wikipedia entry page) and 20 are from Twitter (specified by URL of Twitter user's profile page). All the entities are manually selected by the KBA organizers. However, 29 entities are dropped from the official query list because they have excessive volume of matched documents in the streaming corpus and thus insufficient resource for manual assessment. The remaining 141 entities have moderate matched documents in the streaming corpus.

A new KBA Stream Corpus 2013 is released to support both the CCR and SSF tasks. The corpus ranges from October 2011 through January 2013 with 11,948 hours in total, including the content of KBA Stream Corpus 2012 as a subset. Documents in the first 3,551 hours are for training, and the remaining 8,397 hours for testing, respectively. Due to the large volume of data set (4.5 TB compressed text), it is empirically impractical to apply our related entity based methods on it directly given the limited computation resource. To minimize the computational cost, we first construct a much smaller working set by extracting candidate documents from the original data set. The goal is to identify documents which possibly bear some information about a certain target entity. A simple strategy is to check whether the document mentions the target entity. However, this method would miss many relevant documents as entities may have several surface name variations on the Web. For example, one target entity “Benjamin_Bronfman” has at least three different variations: “Benjamin Zachary Bronfman”, “Ben Brewer” and “BZB”. To ensure the high recall and maintain relative low false positive rate, we follow the work by Balog et al. [1] to extract the surface names of each target entity from its profile page. For Wikipedia entities, we refer to the corresponding DBpedia entry page and extract the attributes which are possible name variations from certain fields (e.g., birthName, sameAs, etc.) by leveraging the structure of DBpedia. For Twitter entities, we use the real names of twitter users as name variations. With the collected name variations, the initial working set construction has been transferred to filtering the documents which match the target entity or any of its name variations. Manual inspection is involved thereafter and name variations which have excessive matched documents are removed as it implies they are common phrases (e.g., short acronyms) in the corpus and documents matching them may not to relevant to the target entity. Documents matching these popular name variations are removed from the working set to make sure the false positive rate is reasonably low. At last, we get a working set with 3.9GB compressed text.

3 Related Entity based Framework

Following our previous work [4], we estimate the relevance score between a given target entity E and document d based on the matching of target entity and its related entities:

$$score(d, E) = \alpha \cdot mention(d, E) + \beta \cdot \sum_{e \in R(E)} occ(d, e) \cdot w(E, e), \tag{1}$$

Notations include:

- $mention(d, E)$: a binary function indicating whether the document d mentions E or any of its name variations (1 means mention exists and 0 otherwise).
- $R(E)$: the set of related entities of E.
- $occ(d, e)$: the number of occurrences of related entity e in d.
- $w(E, e)$: the prior weight of e to control the influence. Important entities will be favored and trivial ones will be penalized by controlling the prior weight.
- α and β: coefficients which balance the impacts of two score components.

The underlining intuition is that more occurrences of e in d means higher confidence we can infer that d is relevant to E. By setting a cutoff threshold $\mathcal{T}$, we only return the documents with score above $\mathcal{T}$ as relevant.

We now discuss how each of the score components is estimated. Since we construct the working set based on the criterion that each document d contains mention to the target entity E or any of name variations, $mention(d, E)$ is 1 for all documents in the working set.

The related entity set $R(E)$ can be generated from the profile page of E. For Wikipedia entities, we follow the URL to retrieve target entity’s profile page and collect related entities by parsing the profile page. Related entities are identified and extracted from the outgoing links to other entity profile pages in Wikipedia. For Twitter entities, we do not collect related entities from the profile page and leave $R(E)$ as an empty set, as we found that the connected entries to the target entity (i.e., the followers and people E is following) are not informative to serve as related entities as the following relationship on Twitter is a weak and non-reciprocal social relation. We plan to explore other ways of collecting related entities for Twitter entities in the future work.

By leveraging the labelled relevant documents $\mathcal{D}_{rel}$ in the training data, we estimate the prior weight of related entity e with regard to E (i.e., $w(E, e)$) in an iterative approach, as described in Algorithm 1. Basically it aims to maximize the performance gain in a greedy way. More details about Algorithm 1 can be found in our previous work [4]. For $w(e^*, R_{sec}(E))$ in line 15, we choose the linear decaying weight (shown in Equation 2) as it shows superior effectiveness over uniform weight.

Algorithm 1 Related Entity Weighting

```
Input: topic entity $E$, related entity set $R(E)$, labeled relevant document set $\mathcal{D}_{rel}$
/*Initialization*/
for $e \in R(E)$ do
    $w(E, e) = 1$
end for
$R_{sel}(E) \leftarrow \{\}$
$R_{left}(E) \leftarrow R(E)$
$G = 0$
while $R_{left}(E) \neq \emptyset$ do
    /* Select the entity which maximizes performance gain when added to $R_{sel}(E)$ */
    $e^* = \arg\max_{e \in R_{left}(E)} Gain(\mathcal{D}_{rel}, R_{sel}(E) \cup \{e\})$
    $G' = Gain(\mathcal{D}_{rel}, R_{sel}(E) \cup \{e^*\})$
    $\Delta G = G' - G$
    if $\Delta G > 0$ then
        /* Re-estimate the weight */
        $w(E, e) = weight(e^*, R_{sel}(E))$
        /* Incrementally augment $R_{sel}(E)$ */
        $R_{sel}(E) = R_{sel}(E) \cup \{e^*\}$
        $R_{left}(E) = R_{left}(E) \setminus \{e^*\}$
        $G = G'$
    else
        /* No further performance improvement */
        for $e \in R_{left}(E)$ do
            $w(E, e) = 0$
        end for
        $R_{left}(E) \leftarrow \emptyset$ /* Force quit the loop */
    end if
end while
```

$$weight(e^*, R_{sel}(E)) = |R(E)| - |R_{sel}(E)| \tag{2}$$

4 Experiment Results

4.1 Submitted Runs

We submitted five official runs to the KBA track. The differences between them are α, β and $w(E, e)$ in Equation (1) and filtering threshold $\mathcal{T}$. Details are summarized as follows.

1. **UDInfoK_EX**: $\alpha = 1000$, $\beta = 0$ and $\mathcal{T} = 0$. It essentially returns all the documents in the working set, in which each document has exact match of the topic entity or any of its name variations. It serves as a baseline.

2. **UDInfoK_Wiki1**: $\alpha = 0$, $\beta = 50$, $\mathcal{T} = 1$ and $w(E, e) = 1$ for all related entities, i.e., uniform weight prior. We drop $mention(d, E)$ as it does not affect the results on the working set. Related entities are utilized as the pivot to estimate the relevance between topic entity and documents. $\mathcal{T}$ is set empirically based on the results of training data. Note that Twitter entities do not have related entities, and we directly return all associated documents in the working set, as what we do in **UDInfoK_Ex**.

3. **UDInfoK_Wiki2**: $\alpha = 0$, $\beta = 50$, $\mathcal{T} = 51$. $w(E, e) = 1$ for all related entities.

Evaluation Set	Vital					Vital+Useful				
Run	mP	mR	mF1	hF1	mSU	mP	mR	mF1	hF1	mSU
all-mean	-	-	-	0.166	0.137	-	-	-	0.376	0.425
all-median	-	-	-	0.174	**0.255**	-	-	-	0.406	0.423
UDInfoK_Ex	0.175	0.675	**0.229**	**0.277**	0.149	0.512	0.795	**0.552**	**0.623**	0.509
UDInfoK_Wiki1	0.174	0.601	0.216	0.269	0.158	0.556	0.708	0.534	**0.623**	**0.515**
UDInfoK_Wiki2	0.174	0.564	0.212	0.265	0.158	0.566	0.662	0.519	0.604	0.510
UDInfoK_Weight1	0.173	0.579	0.207	0.266	0.160	0.556	0.686	0.518	0.614	0.504
UDInfoK_Weight2	0.172	0.563	0.210	0.264	0.157	0.557	0.681	0.524	0.613	0.511

Table 1: Results of official runs. **all-mean** and **all-median** are the mean and median of results aggregated from all the submitted runs in this year's KBA track respectively.

Evaluation Set	Vital					Vital+Useful				
Run	mP	mR	mF1	hF1	mSU	mP	mR	mF1	hF1	mSU
UDInfoK_EX	0.175	0.675	0.229	0.277	0.149	0.521	0.795	0.552	0.623	0.509
UDInfoK_Wiki1	0.243	0.602	**0.259**	**0.346**	**0.215**	0.663	0.708	**0.571**	**0.684**	**0.584**
UDInfoK_Wiki2	0.239	0.564	0.250	0.335	0.213	0.659	0.662	0.548	0.660	0.567
UDInfoK_Weight1	0.241	0.579	0.249	0.341	**0.215**	0.661	0.686	0.554	0.673	0.574
UDInfoK_Weight2	0.233	0.563	0.249	0.330	0.213	0.659	0.681	0.559	0.670	0.576

Table 2: Results of official runs using per-topic cutoff.

4. **UDInfoK_Weight1**: $\alpha = 0$, $\beta = 50$, $\mathcal{T} = 1$ and $w(E, e)$ is estimated by Algorithm 1 and the performance gain function is estimated by harmonic mean of macro-average precision and recall (which is the major evaluation metric in TREC KBA 2012 [2]) over the Vital evaluation set of training data. Note that Algorithm 1 would yield zero weighting for all related entities as none of the related entities would bring performance improvement on the training data. In that case, we use the results of **UDInfoK_Wiki1** instead.

5. **UDInfoK_Weight1**: $\alpha = 0$, $\beta = 50$, $\mathcal{T} = 1$ and $w(E, e)$ is estimated by Algorithm 1 and the performance gain function is estimated by harmonic mean of macro-average precision and recall over the Vital+Useful evaluation set of training data. Topics do not work with Algorithm 1 fall back to **UDInfoK_Wiki1**.

4.2 Results Analysis

The results of all the runs are summarized in Table 1. The mP, mR, mF1, and mSu are the macro-average of precision, recall, F1, and scaled utility over all the queries with a global cutoff, respectively. The hF1 is the harmonic mean of mP and mR, which is used as one of the major evaluation metric in the evaluation. We observe that all of our five runs can reach good results among all the submitted runs. It is interesting to see that our baseline **UDInfoK_Ex** can already deliver satisfying results, implying the name variation based approach is empirically effective on selecting candidate documents. However, runs with related entities involved do not show advantage over the exact matching baseline **UDInfoK_Ex**, actually they even perform worse. Results analysis shows that there are 116 Wikipedia entities with related entities, and the number of related entities for each topic fall in a wide range, with minimum 5, maximum 130, mean 32.647 and standard deviation 22.168, which implies the optimal cutoff for each topic may also fall in a wide range. To verify our hypothesis, we conduct the evaluation using per-topic cutoff, and summarize the results in Table 2. Clearly, methods involving related entities demonstrate advantage over the baseline **UDInfoK_Ex** in terms of both F1 and SU under the per-topic cutoff evaluation setting.

To better understand the performance, we conduct topic-level analysis and plot the per-topic difference between **UDInfoK_Ex** and **UDInfoK_Wiki1** on both Vital and Vital+Useful evaluation

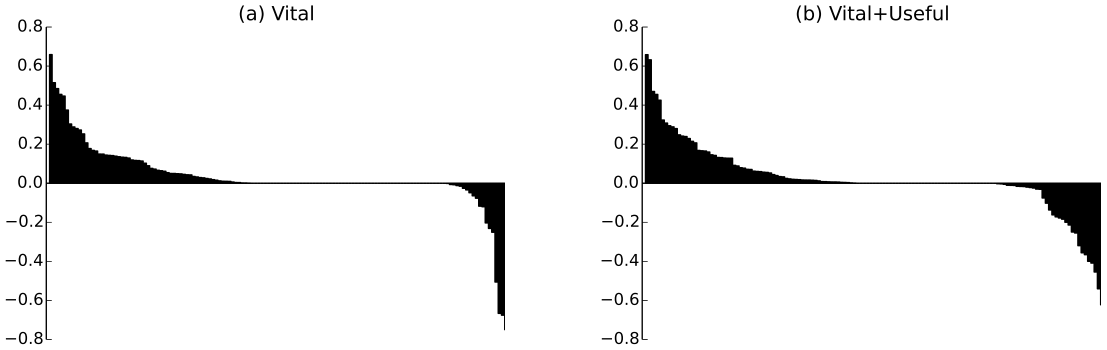

Figure 1: Performance comparison between **UDInfoK_EX** and **UDInfoK_Wiki1** using per-topic cutoff.

settings, as shown in Figure 1. Topics are sorted in the decreasing order of mF1 score differences, positive values mean **UDInfoK_Wiki1** performs better while negative values mean it performs worse. We find that incorporating related entities can improve more topics while hurting less.

We notice that incorporating related entity weighting could not improve the performance over the uniform weight prior, which is different from the observation in our previous work [4]. We conduct result analysis and find that only a few topics can be applied with Algorithm 1. For **UDInfoK_Weight1**, only 5 topics qualify and for **UDInfoK_Weight2**, only 12 topics qualify. We think the main reason is that the topics in 2013 is less popular than those in 2012 and their Wikipedia profile entry is less populated thus have less related entities. Actually the mean of number of related entities in 2012 is 135.897, comparing to 32.647 in 2013. The lack of related entities limits the potential of our approach, and we plan to address it in our future work.

5 Conclusion

This is the second year for us to participate the TREC KBA Track. We extend our approach from last year and evaluate it on the new data set. Experiment results demonstrate that our approach can still deliver good performance under the per-topic cutoff evaluation setting. However, the new data set imposes a new challenge from less populated entity profile, and our approach could not work well. We plan to investigate how to deal with the data sparsity of topic entity profile in our future work.

References

[1] K. Balog, N. Takhirov, H. Ramampiaro, and K. Nørvåg. Multi-step Classification Approaches to Cumulative Citation Recommendation. In *Open research Areas in Information Retrieval (OAIR 2013)*, pages 121–128, 2013.

[2] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. In *Proceedings of TREC*, 2012.

[3] X. Liu and H. Fang. Entity Profile based Approach in Automatic Knowledge Finding. In *Proceedings of TREC*, 2012.

[4] X. Liu and H. Fang. Leveraging Related Entities for Knowledge Base Accleleration. In *Proceedings of 4th International Workshop on Web-scale Knowledge Representation, Retrieval and Reasoning (Web-KR)*, 2013.