WHU at TREC KBA Vital Filtering Track 2014

Chuan Wu
School of Information Management
Wuhan University
Wuhan, Hubei, China
wu.chuan@whu.edu.cn

Wei Lu
School of Information Management
Wuhan University
Wuhan, Hubei, China
reedwhu@gmail.com

Pengcheng Zhou
School of Information Management
Wuhan University
Wuhan, Hubei, China
pc.zhou@whu.edu.cn

Xiaohua Feng
School of Information Management
Wuhan University
Wuhan, Hubei, China
fxh9009@gmail.com

ABSTRACT
This paper describes the WHU IRLAB participation to the Vital Filtering task of the TREC 2014 Knowledge Base Acceleration Track. In this task, we implemented a system to detect vital documents that could be used for a human editor to update or create the profile of an entity. Our approach is to view the problem as a classification problem and use Stanford NLP Toolkit to extract necessary information. Various kinds of features are leveraged to classify documents to three classes, i.e. vital, useful, and non-useful (garbage or neutral). We submitted four runs using different combinations of features. The results are presented and discussed.

1. INTRODUCTION
Knowledge Bases (KBs) have been widely used in many applications, such as entity retrieval, entity linking. With the rapid development of the world and extensive newly generated knowledge, it becomes critical to keep KBs up-to-date. However, many KBs, such as Wikipedia, are maintained manually by human editors. It is time-consuming for human editors to find relevant information of an entity and fill it into entity profile in KBs. If we can provide relevant information of an entity automatically to human editors, much time and effort could be saved to leave human editors to focus on summarizing relevant information and updating KBs.

This problem was addressed by Vital Filtering task in Text Retrieval Conference (TREC) Knowledge Base Acceleration (KBA) track in 2014. As the task is the continuous of Cumulative Citation Recommendation (CCR) task in TREC KBA 2012 and 2013, we refer this task as CCR in this paper. The target of a CCR system is to find candidate relevant documents for a given set of entities from the given chronological stream corpus. The target entity set includes 71 entities, 4 of which are not considered in evaluation because its lack of vital documents. In the remaining 67 entities, 23 entities are from Wikipedia. All entities are labeled with one of three types: Facility (FAC), Person (PERSON), and Organization (ORG).

The stream corpus used in KBA 2014 is a superset of that of KBA 2013, ranging from October 2011 to May 2013. As the size of the streamcorpus is massive, which may lead KBA into an engineering problem rather than a research problem, KBA organizers filtered the corpus from 11TB to 696GB. We submitted 4 runs to KBA CCR Track 2014. Several features are proposed to reveal the relevance between entities and documents.

2. PRE-PROCESSING
2.1 Entity Profile and Related Entities
All target entities for CCR are defined by one or more external profiles. Each external profile is an URL. However, the last modified date of these URLs is unknown. Therefore, we do not use the content of these URLs to avoid violation of the "no future information" principle.

In perspective of entity profile URL, two types of entities can be found in CCR target entities, i.e. entities with a Wikipedia page URL, entities with non Wikipedia profile. For entities with one or more Wikipedia page URL, we use the corresponding Wikipedia page as the profile of the entity. We use the 20110115 version Wikipedia dump, which is the earliest one that satisfies the "no future information" principle we can find. A java library named JWPL[1] is used to process Wikipedia data.

For entities with non Wikipedia profile, we extract entity profiles from the training data of the entity, which is specified by the training time range end (TTRE) of the entity. All data before TTRE of an entity is its training data, and all that after TTRE is its testing data. We first collect all vital documents of an entity as a document collection. Assume the number of vital documents in the collection is N, we extract entity profile and its related entities according to a predefined threshold T as follows:

If N<T, we merge all text in these documents as the profile of the entity, and all entities as related entities;

If N=T, we extract the profile as before, and extract entities occurred in at least two documents as related entities.

If N>T, we select related entities by the occurrence of entities in vital documents. We consider entities that occurred frequently enough as non-occasional entities for the target entity, which can be regarded as related entities. We view entities

[1] http://code.google.com/p/jwpl/

occurred in more than K percent of all vital documents of an entity as its related entity. Then documents constructing an entity's profile is selected using the following principle: the union of related entities in all vital documents selected as profile text must cover all related entities identified in previous step; the number of profile document should be as small as possible.

We empirically set parameter T and K as 3 and 30% respectively.

2.2 Training Data

In previous years, all target entities share the same training time range. However, for many entities, there is no enough training data before the given TTRE. So KBA organizers assign a training time range for each entity according to the principle that 20% of annotated vital documents are before the training time range end. Therefore, all training data are extracted according to the specified TTRE, but not a same cutoff.

3. APPROACH

3.1 Classification

Four types of documents are defined in CCR, including vital, useful, neutral, garbage. We view the CCR problem as a 3-class classification problem by combining garbage and neutral as a single non-useful class. We employ Random Forest classifier in Weka toolkit [2] with default parameter settings.

We train a general classifier for all entities using their training data. Besides, a specific classifier for each specific type, i.e. PER, FAC, ORG is trained using all training data of entities belonging to that type. Therefore, for each document-entity pair, two classification results are calculated. If two results are conflicted, we choose the one with higher confidence score. We submitted four runs in different combination of features. The features are listed in Table 1.

Baseline. One general Random Forest (RF) classifier and four specific classifiers are trained for all entities using document features, entity feature, and document-entity features.

BM_TF. All features used in Baseline are used, with one more temporal feature, namely $PreMention(E, h_{12})$ feature. It is used to compare the result with and without temporal feature.

BM_TF_3. All features in BM_TF are used with two additional temporal features. It's designed to see whether expanding time interval of temporal features is useful.

CUSTOM_TF_FIXED. All features in BM_TF_3 are used. Additional Entity Context Features is also included to see whether these features can help improve performance.

3.2 Features

Wang et al [1] has summarized 5 types of features for CCR, including document features, entity features, document-entity features, temporal features, and citation features. We adopt some of these features with a few modifications. In addition, we explore the Entity Context Features to improve the performance. All features are listed in Table 1.

Document Features. Some features only related to document are used to represent basic characteristics of each document, including document length, publication date and the source of the document. The source of document can be one of 9 sources, including news, social, linking, WEBLOG, arxiv, CLASSIFIED, FORUM, MAINSTREAM_NEWS, MEMETRACKER, and REVIEW.

Entity Features. The number of related entities is the only entity feature. For different entities, we extract related entities using different strategies as described in section 2.1.

Document-Entity Features. All document-entity features are listed in Table 1. They are used to reveal the correlation between a document and the context of an entity. The context of an entity is its profile text extracted in pre-processing step. See section 2.1 for details about entity profile.

Temporal Features. We adopt one temporal feature proposed by Wang et al [1], namely $PreMention(E, h)$, which means the number of entity E mentioned in previous h hours before the timestamp of the document. Another temporal feature is not adopted because no daily page view statistics data is found. However, we enrich feature $PreMention(E, h)$ by changing the time length from 10 hours to 12 hours, 24 hours, and 36 hours. Besides, we regard date or other time-related terms as indicator of new information. We extract window text of all occurrences of an entity in a document as its *local context*. We detect time expressions occurred in *local context* using Stanford Temporal Tagger[2] and then calculate the number of days between the time detected and the timestamp of document. Two levels of time interval are considered, i.e. 7 days and 30 days, leading to two temporal features.

Entity Context Features. We propose several entity level features to reveal the relevance between entity and document, such as New Entity Ratio, Local Entity Similarity, and so on. Entities are extracted from documents by using Stanford NER[3] as it is used in several entity level features. The window size is manually set to 5. We regard the occurrence of new entities, i.e. entities that are not the related entities of the target entity, in entity's local context as a signal of new information about the entity. Thus we define New Entity Ratio of target entity te as follows.

$$g(e, E_{rel}(te)) = \begin{cases} 1 & e \in E_{rel} \\ 0 & otherwise \end{cases} \quad (1)$$

$$NERatio(te, D) = \frac{\sum_{e \in E_{win}(te,D)} g(e, E_{rel}(te))}{|E_{win}(te, D)|} \quad (2)$$

$E_{rel}(te)$ is the set of related entities of entity te, and $E_{win}(te, D)$ is the set of entities found in local context of entity te in document D. Local context is also used to calculate local context similarity between an entity and a document. Two local context similarity, i.e. $LCSim_{cos}(E, D)$ and $LCSim_{jac}(E, D)$ are defined as follows.

$$LCSim_{cos}(E, D) = \frac{v_{lc}(E) * v_D}{|v_{lc}(E)| * |v_D|} \quad (3)$$

$$LCSim_{jac}(E, D) = \frac{v_{lc}(E) * v_D}{|v_{lc}(E)|^2 + |v_D|^2 - v_{lc}(E) * v_D} \quad (4)$$

v_{lc} is the term vector of local context of E and v_D is the term vector of document D. In perspective of context entities, we

[2] http://nlp.stanford.edu/software/sutime.shtml

[3] http://nlp.stanford.edu/software/CRF-NER.shtml

Table 1. Features

Feature	Description
Document Features	
log(length)	log of document length
Source	source of the document, including news, social, arxiv, etc
Date	date-hour timestamp of the document
Entity Features	
$N(E_{rel})$	# of related entities of E in its profile text
Document-Entity Features	
$N(D, E)$	# of occurrences of the target entity E in document D
$N(D, E_p)$	# of occurrences of partial names of E in D
$N(D, E_{rel})$	# of occurrences of related entities of E in D
$FPOS(D, E)$	position of first occurrence of E in D
$FPOS_n(D, E)$	$FPOS(D, E)$ normalized by document length
$FPOS(D, E_p)$	position of first partial name occurrence of E in D
$FPOS_n(D, E_p)$	$FPOS(D, E_p)$ normalized by document length
$LPOS(D, E)$	position of last occurrence of E in D
$LPOS_n(D, E)$	$LPOS(D, E)$ normalized by document length
$LPOS(D, E_p)$	position of last partial name occurrence of E in D
$LPOS_n(D, E_p)$	$LPOS(D, E_p)$ normalized by document length
$Spread(D, E)$	$LPOS(D, E) - FPOS(D, E)$
$Spread_n(D, E)$	$Spread(D, E)$ normalized by document length
$Spread(D, E_p)$	$LPOS(D, E_p) - FPOS(D, E_p)$
$Spread_n(D, E_p)$	$Spread(D, E_p)$ normalized by document length
$Sim_{cos}(D, E)$	cosine similarity between document and entity's profile
$Sim_{jac}(D, E)$	jaccard similarity between document and entity's profile
Temporal Features	
$PreMention(E, h_{12})$	# of times E is mentioned in previous 12 hours before the timestamp of document
$PreMention(E, h_{24})$	# of times E is mentioned in previous 24 hours before the timestamp of document
$PreMention(E, h_{36})$	# of times E is mentioned in previous 36 hours before the timestamp of document
SUCountAWeekBefore	# of temporal terms within one week before the timestamp of document
SUCountAMonthBefore	# of temporal terms within one month before the timestamp of document
Entity Context Features	
$NERatio(E, D)$	the ratio of non-related entities in window text
$LCSim_{cos}(E, D)$	cosine similarity between local contexts of entity in document and document
$LCSim_{jac}(E, D)$	jaccard similarity between local contexts of entity in document and document
$LESim_{cos}(E, D)$	cosine similarity between related entities of E and entities in D
$LESim_{jac}(E, D)$	jaccard similarity between related entities of E and entities in D

Table 2. Results of official runs.

RUN	Vital + Useful		Vital Only	
	max(F(avg(P),avg(R)))	Scaled Utility	max(F(avg(P),avg(R)))	Scaled Utility
baseline	.554	.675	.268	.370
BM_TF	.554	.674	.244	.343
BM_TF_3	.547	.667	.235	.318
CUSTOM_TF_FIXED	.553	.673	.232	.364

extract all entities in document to calculate entity level similarity between an entity and a document.

4. Results and Discussion

We now discuss the results of our system on the CCR task. Two evaluation measures are used by TREC for this task, i.e. average F-score and average Scaled Utility (SU).

In vital and useful detection, our baseline run (Baseline) obtain a reasonable result, while all measures of other runs with temporal features in vital and useful is close to Baseline, which means that the temporal features we selected is not effective in helping recognize useful documents.

In vital detection, the performance of other runs are close to or lower than the Baseline, which indicates that the temporal features we selected is not effective in improving the performance. However, temporal features are shown to be useful in other works [1]. These demonstrate that temporal features should be carefully selected. If invalid temporal features are used, it may have negative effect on the performance.

5. CONCLUSION

We present our method for the CCR task. In our first attempt at this task, we have learned various lessons. We will move to more detailed analysis of different source of text. More work will be done to check for the source of errors occurred in classification process.

6. ACKNOWLEDGMENTS

This work is supported by the National Social Science Fund of China (Grant No. 12&ZD1221) and the National Natural Science Foundation of China (Grant No. 71173164).

7. REFERENCES

[1] Wang, J., et al. BIT and MSRA at TREC KBA CCR Track 2013. in Notebook of the TExt Retrieval Conference. 2013.

[2] Hall, M., et al., The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 2009. 11(1): p. 10-18.