IRIT at TREC Knowledge Base Acceleration 2013: Cumulative Citation Recommendation Task

Rafik Abbes, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem

{abbes, sauvagnat, hernandez, boughanem}@irit.fr,
IRIT
118 route de Narbonne F-31062 Toulouse cedex 9

Abstract. This paper describes the IRIT lab participation to the Cumulative Citation Recommendation task of the TREC 2013 Knowledge Base Acceleration Track.
In this task, we are asked to implement a system which aims to detect "Vital" documents that a human would want to cite when updating the Wikipedia article for the target entity.
Our approach is built on two steps. First, for each topic (entity), we retrieve a set of potential relevant documents containing at least one entity mention. These documents are then classified using a supervised learning algorithm to identify which ones are vital.
We submitted three runs using different combinations of features. Obtained results are presented and discussed.

1 Introduction

The goal of the Knowledge Base Acceleration (KBA) track is to help people enrich and update information about entities [1]. This year, we participated for the first time, in the first task of the KBA track which is called Cumulative Citation Recommendation (CCR). In this task, we are given a list of target entities from Wikipedia and Twitter, and we aim at identifying from the given streamcorpus[1], which stream items (documents) are worth citing when updating the target entity profile (e.g., Wikipedia article).

The streamcorpus is about 4.5TB and just over 500 million documents appearing in the period from October 2011 through February 2013. The corpus is divided into 11948 folders, each one represents one distinct hour. As CCR task requires systems to iterate over the hourly directories of data in chronological order, hour by hour, we indexed each single folder separately using the Lucene Software Library[2]. We indexed only the clean visible form (plain text) of the documents without dealing with HTML tags.

In this task, we are provided with annotations from an early portion of the corpus (from October 2011 through February 2012) as training data. For a given topic (entity), a document can be judged as:

- **Garbage**: If it gives no information about target entity.
- **Neutral**: It is informative but not citable.
- **Useful**: It is useful when building a Knowledge Base entry from scratch, but it does not help to update an already existing Knowledge Base entry.
- **Vital**: It contains a timely information about the entity's current state, actions, or situation. This would be citable when updating a knowledge Base entry.

[1] http://s3.amazonaws.com/aws-publicdatasets/trec/kba/kba-streamcorpus-2013-v0_2_0-english-and-unknown-language/index.html

[2] http://lucene.apache.org/

Our approach is inspired from previous works at TREC KBA 2012 [2][3]. It aims at identifying vital documents. We proceed on two steps: we first retrieve a set of relevant documents for the entity query. Then, these documents go through a supervised algorithm to be classified as garbage, neutral, useful or vital.

In Section 2, we detail our approach. Then, in section 3, we present and discuss results and some perspectives.

2 IRIT Cumulative Citation Recommendation approach

2.1 First Step: Retrieving candidate documents

Queries formulation In the CCR task, we distinguish two types of target entities (topics): Wikipedia topics and Twitter topics that were given in the form of URLs. The first step of our approach consists of retrieving a set of candidate documents for the specified target entity. For that, we need to construct a query text for each topic:

- For Wikipedia topics, we used DBpedia to get entities variants like the work described in [3]. We retrieved all entity predicates containing the pattern "/name" or "#label". Then, the corresponding attributes' values are merged to get the final query text.
- For Twitter topics, we used the Twitter API to get the full name of the entity, and used it as query text.

We wanted to get a high recall in this step, so we did not use quotes to search expressions. For example, for the topic *http://en.wikipedia.org/wiki/Jeff_Tamarkin* we submit *Jeff Tamarkin* as query and not *"Jeff Tamarkin"* with quotes.

Retrieval Model To retrieve and score documents, we used Lucene Formula which is based on the Vector Space Model (VSM). The similarity between document d belonging to a folder f, and a query q composed of terms t is evaluated as follow:

$$Score(q,d) \ = \ coord(q,d) \ . \ \sum_{t \ in \ q} tf(t \ in \ d) \ . \ idf(t)^2$$

Where:

- $coord(q,d)$ is a score factor based on how many of the query terms are found in the specified document
- $tf(t \ in \ d) \ = \ \sqrt{frequency(t,d)}$
- $idf(t) \ = \ 1 \ + \ \log \frac{\#documents \ in \ the \ folder \ f}{\#documents \ in \ the \ folder \ f \ containing \ t \ + \ 1}$

2.2 Second step: Supervised classifier

In this step, we classify the obtained documents from the previous step by applying a supervised learning algorithm using three groups of features described in Table 1. We evaluated some learning algorithms (ZeroR, SVM, Bayesian Network, Random Tree, Random Forest) using the Weka plateform [4] and we kept the Random Forest algorithm which outperforms the others on the training data. The output of the classifier is the corresponding class (garbage, neutral, useful, vital) of each document with regard to the entity with a confidence score between 0 and 1. For the submission runs we kept only "vital" documents with confidence score above 0.7.

$G1$**: Basic features**	
- **Score***	Similarity (VSM) score between document and query
- **Rank**	Rank of the document
$G2$**: Document-Centric related features**	
- **cos_similarity_entity_doc***	Cosine similarity between document and the last version of entity profile (before the timestamp)
- **nb_mention_body**	Number of entity mentions in the document body
- nb_mention_body_0_20*	Number of entity mentions in the first 20% of document body
- nb_mention_body_20_40*	Number of entity mentions in the second 20% of document body
- nb_mention_body_40_60*	Number of entity mentions in the third 20% of document body
- nb_mention_body_60_80*	Number of entity mentions in the fourth 20% of document body
- nb_mention_body_80_100*	Number of entity mentions in the last 20% of document body
- **nb_novelty_words**	Number of words expressing novelty in the document body. We defined a list of 44 novelty words for example, create, invent, new, buzz, novelty, etc. We supposed that vital documents are likely to contain these words.
- title_found	1 if the document has a title, 0 otherwise
- title_mention	1 if the document title mentions the entity, 0 otherwise
$G3$**: Timestamp related features**	
	let t is the timestamp of the current document
- **nbDoc_last24h_scr_gt03***	Number of documents where $score > 0.3$ and $t-timestamp > 24h$
- **nbDoc_last72h_scr_gt03***	Number of documents where $score > 0.3$ and $t-timestamp > 72h$
- **nbDoc_last168h_scr_gt03***	Number of documents where $score > 0.3$ and $t-timestamp > 168h$
- **nb_mention_body_last24h***	Number of entity mentions in the document body where $t - timestamp > 24h$
- **nb_mention_title_last24h***	Number of entity mentions in the document title where $t - timestamp > 24h$
- std_dev_0_mention_24h*	Standard Deviation of entity mentions in documents where $score > 0$ and $t - timestamp > 24h$
- std_dev_0_mention_72h*	Standard Deviation of entity mentions in documents where $score > 0$ and $t - timestamp > 72h$
- std_dev_0_mention_168h*	Standard Deviation of entity mentions in documents where $score > 0$ and $t - timestamp > 168h$
- std_dev_1_mention_24h*	Standard Deviation of entity mentions in documents where $score > 0.1$ and $t - timestamp > 24h$
- std_dev_1_mention_72h*	Standard Deviation of entity mentions in documents where $score > 0.1$ and $t - timestamp > 72h$
- std_dev_1_mention_168h*	Standard Deviation of entity mentions in documents where $score > 0.1$ and $t - timestamp > 168h$
- **std_dev_3_mention_24h***	Standard Deviation of entity mentions in documents where $score > 0.3$ and $t - timestamp > 24h$
- **std_dev_3_mention_72h***	Standard Deviation of entity mentions in documents where $score > 0.3$ and $t - timestamp > 72h$
- **std_dev_3_mention_168h***	Standard Deviation of entity mentions in documents where $score > 0.3$ and $t - timestamp > 168h$

Table 1: Features list divided in 3 groups: Basic features, Document-Centric related features and Timestamp related features. Features with star symbol are inspired from the work described in [2].

2.3 Runs

We submitted 3 runs for the CCR task. Documents retrieved in the first step are the same for all runs. The difference between runs concerns the type of features used in the classifier:

- **sig_irit_1:** We use some representative features from each group $G1$, $G2$ and $G3$ (bold features in table 1).
- **sig_irit_2:** We use all features of groups $G1$ and $G2$.
- **sig_irit_3:** We use all features of groups $G1$ and $G3$.

3 Results

	avg(P)	avg(R)	max(F(avg(P),avg(R)))	Scaled Utility
sig_irit_1	0.121	0.038	0.057	0.252
sig_irit_2	**0.147**	**0.048**	**0.072**	**0.255**
sig_irit_3	0.078	0.031	0,044	0.254
TREC Median			0.174	
TREC Max			0.311	

Table 2: Comparison of submitted runs using macro average measures, (cutoff step=10)

Results show that including features characterizing documents appearing in the same period ($G3$) degrades the system performance. We are currently performing some experiments regarding the precision obtained after the first step of our approach. It appears that searching the exact match of entities names using expressions improves results.

References

1. Trec knowledge base acceleration. `http://trec-kba.org/trec-kba-2013.shtml`.
2. Ludovic Bonnefoy, Vincent Bouvier, and Patrice Bellot. Lsis-lia at trec 2012 knowledge base acceleration. In *The Twenty-First Text REtrieval Conference (TREC 2012) Notebook*, Gaithersburg (USA), november 2012.
3. Samur Araujo, Gebrekirstos Gebremeskel, Jiyin He, Corrado Bosscarino, and Arjen de Vries. Cwi at trec 2012, kba track and session track. *Proceedings of the 21 st Text REtrieval Conference, TREC*, 2013.
4. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.