IRIT at TREC KBA 2014

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

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

Abstract. This paper describes the IRIT lab participation to the *Vital Filtering* task (also known as *Cumulative Citation Recommendation*) of the TREC 2014 Knowledge Base Acceleration Track. This task aims at identifying vital documents containing timely new information that should help a human to update the profile of the target entity (e.g., Wikipedia page of the entity). In this work, we evaluate two factors that could detect vitality. The first one uses a Language Model to learn vitality from a sample of vital documents, and the second leverages the bursts of documents in the stream. Obtained results are presented and discussed.

1 Presentation of the task

The aim of the *Vital Filtering* task[1] is to identify vital documents for a given entity. These documents should help knowledge base editors to update the profile of the entity (e.g. its Wikipedia article). A specially filtered subset of the full TREC 2014 StreamCorpus[2] was provided for use in the 2014 TREC KBA Track. It consists of about 20 million timestamped documents from several sources (News, Social, Forum, Blog, etc.) having a size of 639 GB (compressed). The target topic set is composed of 71 entities including persons, organization and facilities. Each entity E has a training end time (that we denote by $TTR_{end}(E)$). Documents before $TTR_{end}(E)$ can be used as training data and those after are used as test documents.

Document annotations were done as follows: a document is considered as *Vital* if it contains a timely relevant information about the entity, *Useful* if it contains relevant but not timely information about the entity, *Neutral* if it mentions the entity without providing any information about it, and *Garbage* if it does not mention the entity. The official metric for the task is the ***hF1***, i.e. the maximum macro-averaged *F1* measured over all confidence cutoff i ($i \in [0, 1000]$ where 1000 corresponds to the highest level of confidence and 0 corresponds to the level in which all documents are kept).

2 A two-step vitality filtering approach

Entity-centric document filtering methods have been classified into two categories: classification and ranking [1]. Unlike our past participation based on a classification approach [2], we propose this year a ranking based vitality filtering approach which involves two main steps: *filtering* and *scoring*.

[1] http://trec-kba.org/trec-kba-2014/vital-filtering.shtml
[2] http://s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html#kba-streamcorpus-2014-v0_3_0-kba-filtered

2.1 Filtering step

The filtering step can be seen as a way to eliminate non-relevant documents. In this step, for each hour, we select only the **topH** documents that match the full entity name based on the following score:

$$Score(d, E) = \sum_{t \in E} \frac{tf(t, d)}{|d|}$$

Where d is a document, and E represents the full entity name extracted from the given topic url. For example, for the topic *https://kb.diffeo.com/Jeff_Mangum*, *E= Jeff Mangum*. $tf(t, d)$ is the term frequency of term t in document d.

In addition, to reject documents matching the entity but more likely to be spam, we apply the following two filters:

- *An enumeration filter* that rejects documents mentioning the entity only in an abusive list of more than n entities such as E_1, ..., E_t, ..., E_n.
- *A links filter* that rejects documents having more than n hyper-links.

2.2 Scoring step

Documents that pass the filtering step are ranked using two vitality factors: a *Language-Model-based factor* and a *Burst-based factor*.

2.2.1 Estimating vitality with the Language-Model-based factor

We use a Language Model to estimate a vitality model for each entity. With this model, we want to detect *"vital" words* that help identify upcoming vital documents. As vitality is unknown a priori, we leverage the set of training vital documents (before $TTR_{end}(E)$). We believe that we can find some common features between vital documents of the training set and a new vital one. Formally, given an entity E and a sample of $\boldsymbol{n}$ vital documents vd_i, we estimate the entity vitality model θ_{V_E} as follows:

$$P(t|\theta_{V_E}) = \frac{\sum_{i=1}^{n} tf(t, vd_i) df(t)}{\sum_{i=1}^{n} |vd_i|} \tag{1}$$

We note that $P(t|\theta_{V_E})$ is not a strict probability distribution
$tf(t, vd_i)$ is the term frequency of term t in document vd_i
$df(t) = \frac{1}{log(\frac{n+1}{m})}$ is used to boost terms often appearing in vital documents, where m represents the number of vital documents for E containing term t.

The vitality score of a new incoming document d with respect to an entity E is evaluated as follows:

$$Score_{LM}(d, E) = \prod_{t \in top_k(\theta_{V_E})} P(t|\theta_d)^{P(t|\theta_{V_E})} \tag{2}$$

Where $top_k(\theta_{V_E})$ is the set of top k terms in θ_{V_E}, and $P(t|\theta_d)$ is estimated using a Dirichlet Smoothing as follows:

$$P(t|\theta_d) = \frac{tf(t, d) + \mu \frac{tf(t,C)}{\sum_{t' \in C} tf(t',C)}}{|d| + \mu} \tag{3}$$

$tf(t, d)$ is the term frequency of term t in the document d
$tf(t, C)$ is the term frequency of term t in the collection C
C is the reference collection composed from early stream documents before $TTR_{end}(E)$
μ is a smoothing parameter used to avoid null probabilities

2.2.2 Estimating vitality with the burst-based factor

We assume that when new information is published about a given entity, this might lead to an accelerated growth of the number of documents describing this new information. Our idea is to consider that a document is vital regarding an entity E if it appears in a burst of documents that match the entity E. We hypothesize that the higher the number of matching documents is in a short period, the higher the probability of having vital documents will be. We leverage this idea in our *burst-based factor*. Formally, for a new incoming document d, we evaluate its vitality with respect to E as follows:

$$Score_{Burst}(d, E) = (1 - e^{\frac{-x^2}{\sigma}}) * \prod_{t \in E} P(t|\theta_d) \quad (4)$$

$1 - e^{\frac{-x^2}{\sigma}}$ is a cumulative distribution function used to capture the bursts of documents in the stream
x is the number of documents mentioning the entity E in the last $\boldsymbol{w}$ hours preceding the time of d
σ is a parameter reflecting the burst importance
$\prod_{t \in E} P(t|\theta_d)$ is used to prioritize the bursty relevant documents

3 Proposed runs and results

3.1 Proposed runs

We proposed several runs that can be classified into five methods based on the scoring function:

- **VLM** runs : $Score_{Vitality} = Score_{LM}(d, E)$ (Equation 2 in Section 2.2.1)
- **ULM** runs : $Score_{Vitality} = Score_{LM}(d, E)^{(u)}$ (similar than the method in Section 2.2.1, but considering *useful* documents rather than *vital*, to build the Language Model)
- **Burst** runs : $Score_{Vitality} = Score_{Burst}(d, E)$ (Equation 4)
- **Product** runs: $Score_{Vitality}(d, E) = Score_{LM}(d, E) * Score_{Burst}(d, E)$
- **Linear** runs: $Score_{Vitality}(d, E) = \alpha Score_{LM}(d, E) + (1 - \alpha) Score_{Burst}(d, E)$. (We evaluated 3 values of the tuning parameter α: 0.25, 0.5 and 0.75)

In the filtering step, we evaluated 3 values of *topH* (top documents matching the entity per hour) : 10, 50 and 100. In addition, we used two spam filters to reject documents that have more than $n = 30$ links, and/or mentioning the entity only in an abusive enumeration of $n = 30$ entities. Values of n in filters are fixed based on some observations on the last year collection.

To set the parameters of the our factors, we performed a 3-fold cross-validation method using the 2013 KBA topics.

- For the *Language-Model-based factor*, we varied μ between 50 and 1000 (step=50), and $top_k(\theta_{V_E})$ between 5 to 100 (step=5). Optimal parameters are to be μ =200 and $top_k(\theta_{V_E})$ =25.
- For the *Burst-based factor*, we jointly tuned the parameters w and σ by varying σ between 1 and 10 with a step of 1, and considering different numbers of hours for w: 1, 6, 12, 24 *and* 48. Optimal values retained are w =24h and σ =3.

Documents confidence scores are assigned as follows: rank 1 gets a confidence value 1000, rank 2 gets a confidence value 999, etc. Ranks greater than 1000 were assigned a confidence value of 0.

Run	i*	mPrecision@i*	mRecall@i*	F1@i*
VLM, topH=10	34	0.319	0.821	0.459
ULM, topH=10	0	0.306	0.900	0.457
Burst, topH=10	0	0.306	0.900	0.457
Product, topH=10	0	0.306	0.900	0.457
Linear($\alpha \in \{0.25, 0.5, 0.75\}$), topH=10	0	0.306	0.900	0.457
VLM, topH=50	34	**0.322**	0.841	**0.466**
ULM, topH=50	0	0.309	0.936	0.465
Burst, topH=50	0	0.309	0.936	0.465
Product, topH=50	0	0.309	0.936	0.465
Linear($\alpha \in \{0.25, 0.5, 0.75\}$), topH=50	0	0.309	0.936	0.465
Linear($\alpha \in \{0.25, 0.5, 0.75\}$), topH=100	0	0.310	**0.938**	**0.466**

Table 1: Comparison of the different configurations of our approach. i* corresponds to the confidence cut-off in which F1 is maximum. Presented results consider the official 71 KBA entities.

3.2 Results

Table 1 compares our different runs using the official metric of the task (***hF1***). First, we notice that the higher the number of documents considered per hour (*topH*) is, the better is the recall, and thus the *hF1*. In addition, we observe that many runs sharing the same value of *topH* obtain exactly the same results independently of the scoring method used. In fact, the runs sharing the same *topH* deliver the same set of documents but with different ranking. All of these runs have the same behaviour as depicted in Figure 1: when the confidence cutoff decrease (i.e., more documents are kept), we notice a constant increase in recall while precision is more stable. As a result, the macro-averaged F1 (harmonic mean of the macro averaged precision and the macro averaged recall) obtains its maximum in the last confidence cutoff $i* = 0$.

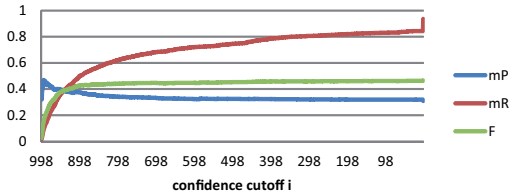

Fig. 1: Linear($\alpha = 0.75$), topH=50 results: macro averaged recall (mR), macro averaged precision (mP) and their harmonic mean F in function of the confidence cutoff i.

In figure 2, we evaluate the impact of our filters (*Enumerations filter* and *Links filter*) when used or not in the VLM, $topH = 50$ run. Considering *None* (VLM, $topH = 50$ run in which no filter is applied) as a reference, we notice that the *Enumerations filter* improves slightly hF1 whereas the use of *Links filters* degrades it slightly. These two filters do not have a significant impact in this year collection, which was not the case on last year collection [3].

In the following paragraph, we investigate a deeper comparison of our configurations.

Deeper comparison of our configurations

We think that our different scoring methods perform differently although they have the same hF1. To verify this, we fixed *topH* (top considered documents in each hour) to 50 (as all our five scoring methods were tested using this value), and we considered only the top 50 documents independently of hours (i.e., $confidence\ cutoff = 950$) (as 50 represents the mean of vital documents for all entities). Results are illustrated in Table 2.

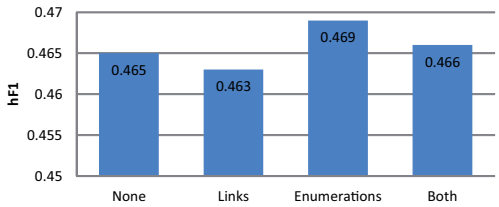

Fig. 2: Impact of each spam filter on hF1 when added to used or not in VLM, $topH = 50$ run

Run	mPrecision@950	mRecall@950	F1@950
Linear($\alpha = 0.75$)	0.396	**0.354**	**0.374**
Product	**0.403**	0.333	0.364
Linear($\alpha = 0.5$)	0.393	0.334	0.361
VLM	0.375	0.341	0.357
Linear($\alpha = 0.25$)	0.385	0.329	0.354
ULM	0.368	0.333	0.350
Burst	0.368	0.333	0.350

Table 2: Comparison of the different configurations of our approach considering the top 50 documents based on confidence score (i.e., confidence cutoff i to 950)

Unlike the hF1 metric, F1@950 (considering the top 50 documents) shows that our runs perform differently. We can therefore report the following finding:

- Results show that VLM run (based on a vital sample of documents) performs better than UVM run (based on a useful sample of documents) which indicates that the *Language-Model-based factor* (described in Section 2.2.1) can learn "vital" that not exist in useful documents. The learned "vital" terms, captured in the training time range, are very helpful to detect upcoming vital documents (in the test time range).
- The *Language-Model-based factor* performs better than the *Burst factor* (0.357 vs. 0.350). The latter factor performs the worst among all configurations.
- Combining both *Language model* and *Burst* factors ($Linear(\alpha = 0.75$ *or* $0.5)$ or $Product$) provides better results than using each factor individually. Furthermore, giving a high weight to the *Language model factor* when using a linear combination ($Linear(\alpha = 0.75)$) provides the best performance.

We can conclude that estimating vitality using a Language Model is promising for vitality detection especially when combined with temporal feature such as the burst. The hF1 metric is unfortunately not an appropriate metric to confirm our findings.

References

1. Krisztian Balog and Heri Ramampiaro. Cumulative citation recommendation: Classification vs. ranking. In *Proc. of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 941–944, New York, USA, 2013.
2. Rafik Abbes, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem. IRIT at TREC knowledge base acceleration 2013: CCR task. In *TREC*, 2013.
3. Rafik Abbes, Karen Pinel-Sauvagnat, Nathalie Hernandez, and Mohand Boughanem. Leveraging temporal expressions to filter vital documents related to an entity. In *ACM Symposium on Applied Computing (SAC) (to appear)*, 2015.