Accelerating the update of knowledge base instances by detecting vital information from a document stream

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

Accelerating the update of knowledge base instances by detecting vital information from a document stream

Rafik Abbes, Nathalie Hernandez, Karen Pinel-Sauvagnat and Mohand Boughanem
IRIT, Université de Toulouse, France
abbes, hernandez, sauvagnat, bougha@irit.fr

***Abstract*—With the growth of Linked Data, updating knowledge bases (KB) is becoming a crucial problem, particularly when representing the knowledge linked to permanently evolving instances. Many approaches have been proposed to extract new knowledge from textual documents in order to update existing KB. These approaches reach maturity but rely on the fact that the adequate corpus is already constructed. In most cases, the considered documents are manually selected which makes an exhaustive update of the KB difficult. In this paper, we propose an original approach to automatically detect from a document stream, pieces of information reporting new knowledge about instances of the KB.**

I. Introduction

Knowledge bases (KB) such as *DBpedia* are now seen as essential to deliver to everyone knowledge about real world instances such as people, organizations, events, ... Knowledge about these instances may evolve over time : people, for example, can carry out new actions, as well as be faced with new situations. This requires a permanent effort to maintain KB up-to-date.

Extracting knowledge from textual documents is a commonly used approach for KB creation [1]. Existing approaches are often based on the assumption that the corpus from which knowledge is extracted is already identified, either from Wikipedia pages in the case of DBpedia, or made manually [2][3]. In the context of updating KB, identifying texts from which to extract the knowledge is not a trivial task. On the one hand, some approaches aiming at extracting knowledge analyse documents in their entirety [2][3], while knowledge is often contained in some sentences. On the other hand, some particular instances such as long-running events (like disasters) evolve quickly over time and texts on which these approaches are applied should provide new and updated information.

In this paper, we propose an approach to identify in a document stream sentences that report timely and relevant information on instances already existing in a KB. These sentences are called *vital sentences*. Detecting such sentences in real-time is a very complex task that can be decomposed in several open issues (steps): First, how to detect if a document reports a vital information on a given instance? Second, how to extract vital sentences? And third, how to detect if two vital sentences report the same knowledge (information redundancy)?

The *first step* requires separate investigation we presented in previous work [4]. In this paper, we make this step easier by analysing only some periods of the stream in which documents mentioning the instance tend to report vital information. The *second step* is crucial since it allows the selection of candidate vital sentences. State-of-the-art works are generally based on the presence of specific words (chosen either manually or automatically) to evaluate the relevance of the sentence. In this work, we propose to exploit the knowledge already present in the KB. We aim at identifying the specific vocabulary for each type of instance. For the third step, we exploit here again the knowledge represented in the KB and we detect the novelty by combining the textual divergence and the recognition of related instances in the sentences.

In section 2, we present related work about the identification of vital information. Section 3 presents our approach to identify vital information based on the exploitation of knowledge about similar instances. The interest of our approach compared to state-of-the-art methods is discussed in section 4. We conclude and give some perspectives in section 5.

II. Related work

Accelerating the update of KB is a recent open issue whose first concern is to identify a need for change. Analysing the content of documents from which to extract the knowledge to update is a solution to identify this need [5]. Identifying documents is often left to the KB designers. However, when the considered KB deals with instances widely mentioned on the Web, it is a shame not to take advantage of this information. For instance, the *DBpedia Live* aims at updating the KB in near real-time when Wikpedia's pages infoboxes are modified [6]. However, as highlighted in [7], some delay can be observed on the update of Wikipedia pages although the information is published in real-time on the web. To face this problem, a recent track called *Knowledge Base Acceleration (KBA)* [8][7] was proposed by the TREC evaluation campaign. Several methods have been proposed [8] to filter vital documents reporting new facts about instances from a document stream. However, these approaches do not deal with redundancy and select whole documents making KB editors or extraction tools deal with a huge amount of content.

Other approaches are interested in the identification of vital sentences concerning well-known events such as natural disasters in a stream of web documents [9][10]. [11] uses training data to learn important terms to identify vital sentences. [12] uses a classifier to detect sentences containing new and prominent information. In [13], sentences are selected according to the frequency of important keywords obtained by using the

Latent Dirichlet allocation algorithm. In this work, we detect vital sentences for a new event by exploiting important terms associated to the knowledge already represented in the KB.

III. Real time detection of vital information related to a given instance

The aim of our approach is to detect from a document stream, sentences reporting new and relevant information related to a given instance of a KB. These vital sentences can be used to update the knowledge about the instance. Consequently, they should be relevant, exhaustive (cover all the different information published on the instance), non-redundant and detected without significant delay.

Formally, let us consider a stream DS composed of documents d having a publication date $t(d)$ and a sequence of sentences s_j such as $0 \leq j < l(d)$ where $l(d)$ is the document d length, i.e. the number of sentences it contains. Let $h_0, h_1, \ldots, h_n$ be instants separated by a constant time interval (one hour for instance). DS_{h_i} is the set of documents in the stream such as $\forall d \in DS_{h_i}, h_{i-1} \leq t(d) < h_i$.

Let $S(I)$ be the set of vital sentences to be detected from the stream DS (Initially, $S(I) \leftarrow \{\}$). Our approach works iteratively: For each instant h_i, we distinguish 3 main steps : (1) selection of vital documents VD_{h_i} for a given instance I, by using the instance's label in the KB as a query, (2) selection of candidate vital sentences, and (3) verification of the novelty of candidate sentences. Novel sentences are then added to the set of vital sentences $S(I)$.

A. Vital document selection

We only analyse the "hot period" during which vital information on a given instance is published in the stream. At each instant h_i, we analyse new documents appearing in the stream between h_{i-1} and h_i and we assign a vitality score to each document-instance pair. This score is evaluated by the probability that terms composing the instance label are generated by a language model estimated from the considered document [14]. The ***top-h*** of documents are selected to be analysed in the next step.

B. Vital sentence selection

In this step, we analyse the sentences of the selected documents. For each sentence, we have to decide whether it is vital or not to the instance I. We rely on the following intuitions: a vital sentence

- is close to an occurrence of I (i.e. close to an occurrence of the term(s) of I labels),
- contains "important" terms relative to I.

The proximity of a sentence with respect to I can reflect its relevance. A sentence mentioning the instance is more likely to be related to it. We express the proximity of sentence s_j with instance I having the label $I.l$ using the following equation:

$$proximityScr(s_j, I) = \frac{1}{|I.l|} \sum_{t \in I.l} \sum_{d=0}^{dmax} e^{-d \, * \, match(t, s_{j+d}, s_{j-d})}$$

$|I.l|$ is the number of terms in $I.l$, $match(t, s_x, s_y)$ is equal to 1 if t is contained in one of the sentences s_x and s_y, 0 otherwise, and $dmax$ is the maximal distance to consider (in number of sentences). We consider only the sentences in proximity to I by favoring those close to all of the terms composing the instance label, i.e, having a *proximityScr* $> \tau_p$. For example, if the instance considered is labelled with the terms "Hurricane Sandy", sentences in proximity with both terms will be boosted.

In addition to proximity, we consider that each instance I can be associated to a set of important terms that would help reflect sentence vitality. We call these terms *trigger terms*. As we make a real-time analysis, we do not know a priori what are the terms that could reflect the vitality of a sentence. We thus make the following hypothesis: similar instances (i.e. instances of the same type) might share the same trigger terms. To identify these terms, we propose to exploit all the annotations (description of instances in natural language) that have been represented the instances that share the same type as that of the considered instance. We consider as annotations the strings that have been associated to an instance by any annotation properties of OWL (rdfs:comment, rdfs:seeAlso, ...) or KB specific annotation properties such as DBPedia *dbpedia-owl:abstract*. Our idea is thus to exploit descriptions containing as many details as possible to extract trigger terms. For instance, terms such as *effects*, *force*, *storm*, *injuries*, *damage* might be very useful to describe instances of the same type as *Sandy hurricane* or *Isaac hurricane*.

Formally, let $X(I) = \{A(I_1), A(I_2), \ldots, A(I_m)\}$ be the set of the m annotation property values associated to instances of the same type as I. We weight terms t according to the following equation:

$$\omega(t) = \frac{\sum_{i=1}^{m} TF(t, A(I_i))}{IIF(t, I)} \tag{2}$$

$TF(t, A(I_i))$ is the number of occurrences of term t in annotation $A(I_i)$, $IIF(t) = log(\frac{m+1}{IF(t)})$ is a factor used to give priority to terms that are in most of the annotations and $IF(t)$ is the number of instances in the type whose annotation contains term t ($IF(t) \leq m$). The ***top-k*** terms will be considered as trigger terms for instance I.

C. Novelty detection

Sentences that were selected in the previous step could contain redundant vital information (i.e. vital information that has already be identified). To remove redundancy, we compare each candidate vital sentence to all vital sentences already in the incremental set $S(I)$. Detecting novelty is not an easy task. Two sentences may have many terms in common, but report two different information, and inversely they may be divergent but contain the same information.

In our approach, we consider that a candidate vital sentence s_j is novel with regard to already issued sentences ($S(I)$) if its text is divergent (**DIV**) and/or contains New Related Instances (**NRI**), not detected in the preceding sentences $S(I)$. Formally s_j is novel if it fulfills the following conditions:

$$is\_novel(s_j, S(I)) = DIV(s_j, S(I)) \circ NRI(s_j, S(I)) \tag{3}$$

(1)

$$DIV(s_j, S(I)) = \begin{cases} \text{false} & if \exists s_k \in S(I), cos(s_j, s_k) > \tau_n(S(I)) \\ \text{right} & otherwise \end{cases} \tag{4}$$

$$NRI(s_j, S(I)) = \begin{cases} \text{right} & if \exists x \in RI(s_j, I), \forall s_k \in S(I)\ x \notin RI(s_k, I) \\ \text{false} & otherwise \end{cases} \tag{5}$$

$RI(s_k, I)$ is the set of related instances recognized in sentence s_i and potentially semantically linked to I. We propose to take into account the object and data properties

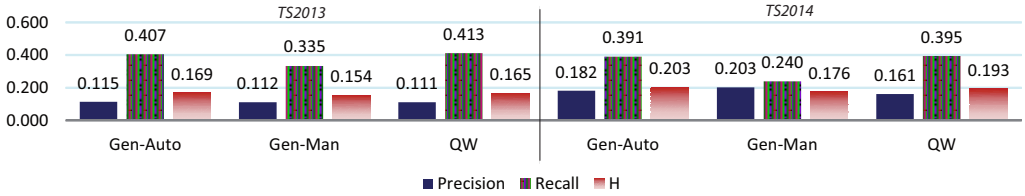

Fig. 1: Comparison of the different strategies for trigger term selection (Gen-Auto, Gen-Man, QW)

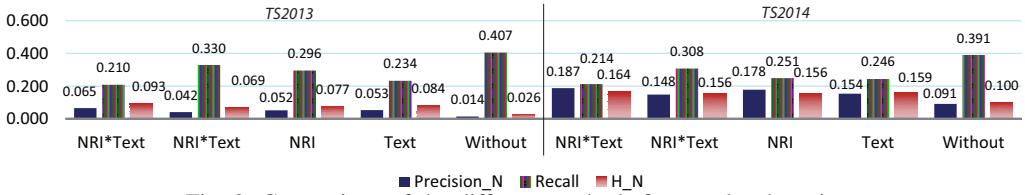

Fig. 2: Comparison of the different methods for novelty detection

defined in the ontology for the type concept of the considered instance. We want to identify in the sentence, instances or values that are semantically linked to the considered instance.

$\tau_n(S(I))$ is a threshold for textual novelty. As the set of vital sentences $S(I)$ grows, the redundancy risk is higher. We thus decrease τ_n according to a Gaussian function $\tau_n(S(I)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{|S(I)|^2}{\delta^2}}$ where the σ parameter has an impact on similarity tolerance, and the δ one controls the decay rate of the threshold. $|S(I)|$ is the number of sentences in $S(I)$.

The $\circ$ symbol of equation 3 can either be an ***AND*** operator to tune the system as precision-oriented by limiting redundancy or an ***OR*** operator to prioritize the exhaustivity.

IV. Experiments

We evaluated our approach within the Temporal Summarization task of TREC which aims to develop systems that are able to monitor events by detecting all new information about them from a data stream containing 500 million English web documents (News, blogs, etc.). These documents were published in the time range of October 2011 through April 2013. Systems should extract sentences containing vital information while avoiding redundancy. 24 topics[1] are proposed by the task organizers in 2013 and 2014. They correspond to events (such as protests, accidents or natural disasters) and can be mapped to DBpedia instances for which we would like to identify vital sentences. Vital information to be retrieved are extracted from different updates of the Wikipedia pages of these events. We thus consider that it should also have led to a manual update of DBpedia. A sentence is judged vital if it can be associated to at least one vital information. We use the usual metrics of Recall and Precision to evaluate the capacity of a system to return vital sentences, without penalizing redundancy. To consider novelty (i.e. penalize redundancy), we used a modified function of precision by considering a redundant sentence as a not relevant.

A. System configuration

To select candidate vital sentences, we apply the method described in section III-B which is based on the detection of the ***top-k*** trigger terms by leveraging the annotations associated to instances having the same type as the considered instance (Eq. 2). We call this strategy as ***Gen-Auto***. We evaluate two other strategies: ***Gen-Man*** in which we manually select 15 generic terms that can characterize most of the considered events. Among these terms, we can cite: *dead*, *death*, *kill*, *injuries*, *damage*, etc. The third strategy ***QW*** simply uses keyword terms of the label as trigger terms.

To detect the novelty of candidates sentences comparing the previously selected ones, we evaluate the following methods: ***Text***: using only textual novelty (Eq. 4), ***NRI***: using only the detection of related instances (Eq. 5), ***NRI*Text*** (***NRI+Text***) using the novelty detection function combined with an AND (OR) operator respectively (Eq. 3), and ***Without*** by considering the selected sentences in step 2.

We fixed our system parameters using cross-validation. The resulting optimal values are the following: *top-h=10*, *top-k=15*, $\tau_p = 0.8$, $\delta = 200$ and $\sigma = 0.5$.

B. Results analysis

After the first step, our system returns 20800 documents for 24 instances with an average recall of 0.65.

1) Strategies for selecting trigger terms: Fig. 1 compares the different strategies of trigger terms selection for detecting vital sentences without taking into account redundancy (***Without***). Considering only the label keywords (***QW***) allows the capture of almost **63%** (0.407/0.650) of vital information contained in selected documents with a precision that does not exceed ***0.161***. The proximity condition (Eq. 1) with a threshold $\tau_p = 0.8$ seems to be strict since it requires the presence of the majority of the label keywords in the sentences. This can explain the loss of 37% of vital information. The use of **Gen-Auto** is similar to the verification of the presence of the label keywords and a generic term at the same time. As a result, we notice a slight improvement of precision with regards to *QW* practically without losing recall. The recall stability shows that prominent terms automatically extracted from the annotation properties of similar instances allow to cover the different aspects of the new instance. The increase of the precision proves the importance of these terms. The manual selection of generic keywords (**Gen-Man**) increases recall (especially for 2014 instances) but still relatively lower comparing to the use of the automatic method **Gen-Auto**.

2) Comparison of the different configurations for novelty detection: Fig. 2 compares different configuration for novelty detection. Detecting novelty increases $precision_N$ and penalizes recall. Combining the textual divergence with the related instance recognition ***NRI*Text*** obtains a better harmonic mean

[1]Topics are available at *www.trec-ts.org/documents*

TS 2013				TS 2014			
System	**ELG**	**LC**	**H-TS**	**System**	**ELG**	**LC**	**H-TS**
*Gen-Auto; Text*NRI*	0.1102	0.1986	**0.1355**	cunlp	0.0631	0.322	**0.1162**
Gen-Auto; Text+NRI	0.0768	0.2619	0.1188	BJUT	0.0657	0.4088	0.1110
ICTNET	0.0794	0.3636	0.1078	*Gen-Auto; Text*NRI*	0.0881	0.1646	0.1047
PRIS	0.136	0.195	0.1029	uogTr	0.0467	0.4453	0.0986
HLTCOE	0.0522	0.2834	0.0827	*Gen-Auto; Text+NRI*	0.0712	0.2181	0.0963

TABLE I: Comparison of our system with the official participants to the TS 2013 and 2014 tasks. H-TS is the harmonic mean between ELG and LC, used as the official metric.

Instance	Vital information detected	t_{web}	t_{wp}	t_{IB}	*Gain in hours* (t_{wp} - t_{web})
1	*550 injured*	22-02-12 16:05	22-02-12 22:49	22-02-12 22:49	6.7
1	*crashed at speed of 26 kilometers per hour*	22-02-12 22:21	22-02-12 23:01	Not available	0.67
9	*39 casualties reported in Guatamala*	08-11-12 00:34	08-11-12 04:33	08-11-12 04:33	3.98
9	*48 casualties reported*	08-11-12 07:42	08-11-12 07:55	08-11-12 07:55	0.22
19	*over 5000 people in the streets of Romanian cities*	16-01-12 03:58	18-01-12 02:28	Not available	46.5
19	*Queensland floods*	27-01-13 11:35	24-01-13 22:42	Not available	60.8

TABLE II: Some examples of vital information detected by our approach (Gen-Auto, NRI*Text). t_{web}, t_{wp}, t_{IB} are the times at which information was published/detected by respectively our system, Wikipedia and Wikipedia infoboxes.

(H_N) between $recall$ and $precision_N$ for 2013 and 2014 instances. Using the *NRI+Text* strategy is useful if we favour knowledge exhaustivity for the instance.

3) Comparison of our system with the task participants: Table I compares our system with the task participants using the official evaluation tool[2]. Evaluation metrics are *ELG* and *LC* which are similar to precision and recall respectively but penalize redundancy and latency when emitting sentences [9]. Systems are ranked using the *H-TS* : the harmonic mean between *ELG* and *LC*. Our system would have been ranked first (/7) for the TS 2013 task, and third (/6) in 2014. Our system therefore appears to be effective for the detection of vital sentences for updating KB and it can be tuned for precision (*Gen-Auto; Text*NRI*) or for recall (*Gen-Auto; Text+NRI*).

4) Rapidity of our approach with regard to Wikipedia updates: We compared our system rapidity (Gen-Auto; NRI*Text) to detect vital information for the 24 events with regard to Wikipedia updates. Our system detects 67% of vital information before it is updated in Wikipedia. On average, our system gains 18 hours. Table II show examples of information published in web documents before being edited in Wikipedia. Although analysed instances related to well-known events are the concern of several contributors, we notice a latency time for Wikipedia updates compared to web documents. This latency time should be bigger for less known instances. Note that the update is not necessarily reported in the page Infobox which is mainly exploited for updating DBpedia. Designing new methods that analyse web documents in real-time in order to accelerate KB updates is thus in our opinion crucial.

V. Conclusion

In this work, we proposed a method that extract vital information as it is published on the web. Such approach is useful not only to help update of documents describing instances like wikipedia pages, but also for updating KB entries themselves because they help identifying specific sentences that can be analysed by extractors. The experiment we conducted shows that finer updates of DBpedia could be performed by identifying from the web real-time vital sentences containing information that is not reported in the infobox. In the short term, we intend to continue the evaluation of our system by using knowledge extraction tools on the detected vital sentences.

[2]http://www.trec-ts.org/documents

References

[1] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos, "Ontology population and enrichment: State of the art," in *Knowledge-driven multimedia information extraction and ontology evolution*. Springer-Verlag, 2011, pp. 134–166.

[2] I. Augenstein, S. Padó, and S. Rudolph, "Lodifier: Generating linked data from unstructured text," in *Proceedings of the 9th International Conference on The Semantic Web: Research and Applications*, ser. ESWC'12, Berlin, Heidelberg, 2012, pp. 210–224.

[3] P. Exner and P. Nugues, "Entity extraction: From unstructured text to dbpedia rdf triples," in *Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web Conference (ISWC 2012)*. CEUR-WS, 2012, pp. 58–69.

[4] R. Abbes, K. Pinel-Sauvagnat, N. Hernandez, and M. Boughanem, "Leveraging temporal expressions to filter vital documents related to an entity," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. Salamanca, Spain: ACM, 2015, pp. 1093–1098.

[5] F. Zablith, G. Antoniou, M. d'Aquin, G. Flouris, H. Kondylakis, E. Motta, D. Plexousakis, and M. Sabou, "Ontology evolution: a process-centric survey," *The Knowledge Engineering Review*, vol. 30, no. 01, pp. 45–75, 2015.

[6] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web Journal*, 2014.

[7] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Re, and I. Soboroff, "Building an Entity-Centric stream filtering test collection for TREC 2012," in *Proceedings of the Text REtrieval Conference (TREC)*, 2012.

[8] J. R. Frank, S. J. Bauer, M. Kleiman-Weiner, D. A. Roberts, N. Tripuraneni, C. Zhang, and C. Re, "Evaluating stream filtering for entity profile updates for trec 2013," in *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA., 2013.

[9] J. Aslam, F. Diaz, M. Ekstrand-Abueg, V. Pavlu, and T. Sakai, "Trec 2013 temporal summarization," in *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, USA., 2013.

[10] R. McCreadie, C. Macdonald, and I. Ounis, "Incremental update summarization: Adaptive sentence selection based on prevalence and novelty," in *Proceedings of International Conference on Information and Knowledge Management*, New York, 2014, pp. 301–310.

[11] Q. Liu, Y. Liu, D. Wu, and C. Xueqi, "Ictnet at temporal summarization track trec 2013," in *Proceedings of the Twenty-Second Text REtrieval Conference*, Gaithersburgh, MD, USA., 2013.

[12] T. Xu, P. McNamee, and D. W.Qard, "HLTCOE at TREC 2013: Temporal summarization," in *Proceedings of the Text REtrieval Conference*, Gaithersburgh, 2013.

[13] C. Zhang, W. Xu, F. Meng, H. Li, T. Wu, and L. Xu, "The information extracion systems of pris at temporal summarization track," in *Proceedings of the Text REtrieval Conference*, Gaithersburgh, 2013.

[14] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st Annual International ACM SIGIR Conference*, New York, NY, USA, 1998, pp. 275–281.