USING TF-ISF WITH LOCAL CONTEXT TO GENERATE AN OWL DOCUMENT REPRESENTATION FOR SENTENCE RETRIEVAL

Alen Doko[1], Maja Štula[2] and Ljiljana Šerić[3]

[1]JP Croatian Telecommunications d.o.o., Mostar, Bosnia and Herzegovina

[2]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia

[3]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, Croatia

ABSTRACT

In this paper we combine our previous research in the field of Semantic web, especially ontology learning and population with Sentence retrieval. To do this we developed a new approach to sentence retrieval modifying our previous TF-ISF method which uses local context information to take into account only document level information. This is quite a new approach to sentence retrieval, presented for the first time in this paper and also compared to the existing methods that use information from whole document collection. Using this approach and developed methods for sentence retrieval on a document level it is possible to assess the relevance of a sentence by using only the information from the retrieved sentence's document and to define a document level OWL representation for sentence retrieval that can be automatically populated. In this way the idea of Semantic Web through automatic and semi-automatic extraction of additional information from existing web resources is supported. Additional information is formatted in OWL document containing document sentence relevance for sentence retrieval.

KEYWORDS

Sentence Retrieval, TF-ISF, Context, Document Level, OWL, Document Representation

1. INTRODUCTION

Sentence retrieval is the task of finding relevant sentences from a document base in response to a query. Tasks like novelty detection [1-5] question answering [2, 6], text summarization [7] and information provenance [2] make use of sentence retrieval.

In the scope of this paper our focus is on sentence retrieval for Novelty detection which deals with finding relevant and at the same time new sentences. The main reason for choosing Novelty detection is that we have test collections for it where each sentence is labeled as relevant or non-relevant by a human assessor. Sentence retrieval methods are usually simple adaptations of document retrieval methods where sentences are treated as documents [3-5]. One of the first and most successful methods for sentence retrieval is the TF-ISF method [8] which is an adaptation of the TF-IDF method [9] to sentence retrieval.

TF-ISF method is based on vector space model of information retrieval. Sentence retrieval based on the vector space model is illustrated in Figure 1. Each sentence in the document collection is represented as a vector. A query is also represented as a vector. The vector space is defined based

DOI:10.5121/cseij.2015.5501

on terms appearing in the document collection. In the figure we are presenting only a 3-dimensional subspace, defined by terms t1, t2 and t3, for visualization purposes, but any n-dimensional subspace can be used depending on the searched terms. These terms are also the terms appearing in the query. Sentences whose vectors are most similar to the vector representing question are retrieved. Retrieved sentences are highlighted.

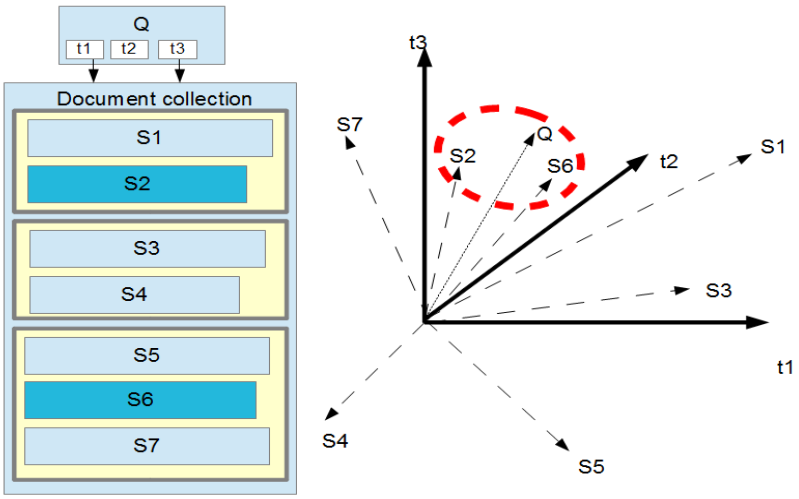

Fig. 1 Illustration of sentence retrieval based on vector space model

The TF-ISF method has outperformed other methods like BM25 based methods or language modeling based methods [8, 10, 11] and that is why we have focused our research to the TF-ISF method. Another, typically used, method for sentence retrieval is the query likelihood method which is a language modeling approach to document retrieval. This method, invented by Ponte & Croft for document retrieval [12] was adapted and often used for sentence retrieval [2]. The TF-ISF and the query likelihood method can be considered baselines methods for sentence retrieval. The TF-ISF method and the query likelihood method were used as the baseline methods for sentence retrieval in [1] and the query likelihood method was used as the baseline [2].

In the recent works [1, 2] query likelihood was modified to take into account the local context of the sentence. Due to the sparsity of sentences there is usually little overlap between the query and the sentence which negatively affects the performance of sentence retrieval [1]. The assumption is that this problem can be partially solved by using the local context of sentences.

The idea that "good" sentences come from "good" documents was proposed by Murdock [2]. So the query likelihood method was improved using local context in the form of the document the sentence came from. A mixture model was proposed combining a sentence language model, document language model and collection language model. The method showed better results when compared to query likelihood baseline [2].

The sentence's local context was also incorporated into the language modeling framework to better estimate the relevance of a sentence [1]. The document that contains the sentence and surrounding sentences (previous, current and the next sentence) were used as the local context. Additionally, the importance of a sentence within a document or $p(d|s)$was used. Tests showed significant improvements of language modeling methods when using local context in comparison to baselines like TF-ISF, and BM25 and to language modeling methods that ignore local context. Including sentence importance additionally improved the performance of tested language modeling approaches.

Fernandez et al. [1] also tried to improve the TF-ISF method by modifying it to take into account the local context. Two types of local context, the document that contains the sentence and surrounding sentences (previous, current and the next sentence) were tested again. They tried to modify the TF part to take into account the number of occurrences of term in the context and also tried to compute the ISF part at document level rather than at sentence level. The tests did not show consistent and significant improvements.

A local context, implemented as a sliding window, was also used by Tsai et al. in [13] for the task of sentence retrieval. A sliding window consisted of multiple sentences. The whole sliding window was compared to the topic and if the sliding window is relevant all sentences inside it are considered relevant. The retrieval method was based on comparing nouns and verbs in the sliding window and the topic. Some tests showed best result when the sliding window is of size 4 but no comparisons were made to the state of the art methods.

The local context of sentences is not the only thing used for improvement of sentence retrieval methods. In [10] Losada et al. extracted highly frequent terms from top retrieved documents and calculated a score that is based on the number of highly frequent words inside the current sentence. That score was combined with the classical TF-ISF score and showed improved results over the state of the art method TF-ISF. The use of pseudo relevance feedback also proofed useful when applying to sentence retrieval [14, 15].

It is important to emphasize that this paper is interdisciplinary. At the one hand it deals with information retrieval and at the other hand it deals with Semantic Web and ontology learning and population which is essential for Semantic Web. Therefore we present here a short introduction into Semantic web. Semantic web can be defined as a web of data that can be processed directly or indirectly by machines [16]. The key element of the Semantic Web are ontologies which are used to add metadata onto the web. Ontology can be defined as an explicit specification of a common conceptualization [17]. Common elements of an ontology are classes, subclasses and relations. The automatic generation of ontologies is called Ontology Learning and the automatic instantiation of ontology elements (e.g. classes, relations) is called Ontology Population. Ontologies are presented using ontology languages like RDF (Resource Description Framework), RDFS (Resource Description Framework Shema) and OWL (Ontology Web Language). Using ontologies for describing the content on the web we create a more intelligent web that can more easily be processed by machines which allows the end users a better user experience. In this paper we show how a new method for sentence retrieval can be converted into an ontology and also how this ontology can be automatically populated. Converting the task of sentence retrieval to standard semantic web technology (OWL) we simplify the development of a sentence retrieval system and can help spreading sentence retrieval. That simplification of different tasks using standard OWL is in the sense of the Semantic Web.

In this paper we at first modify our previous method TF-ISF_{con} to only take into account information on level of document. We empirically test the performance of the new method on datasets from TREC 2002, 2004 and 2004 Novelty Tracks. In Section 3, we show that it is possible to generate a document level OWL representation of a textual document that implies an ontology definition and a way how to automatically populate the ontology. The OWL document representation allows easier development of sentence retrieval systems freeing the system builder from information retrieval specific tasks like preprocessing and focusing on standard semantic web technologies like OWL.

2. DOCUMENT LEVEL TF-ISF AND DOCUMENT LEVEL TF-ISF$_{CON}$

In [18] it was shown that it is possible to improve the TF-ISF ranking function using local context and named the new method TF-ISF$_{con}$. In [18] the method TF-ISFcon was derived from R(s|q) as follows:

The TF-ISF based ranking function for sentence retrieval is [8, 14]:

$$R(s|q) = \sum_{t \in q} \log(tf_{t,q} + 1) \log(tf_{t,s} + 1) \log\left(\frac{n+1}{0.5 + sf_t}\right) \quad (1)$$

Where

- $tf_{t,q}$ is the number of occurences of term t in query q
- $tf_{t,s}$ is the number of occurrences of term t in sentence s
- sf_t number of sentences that contain term t
- n number of sentences in the collection

In [18] the previous and next sentence in the same document were used as the local context of each sentence. It was assumed that relevance of a sentence depends partially on the information within the sentence itself and partially on information within the two closest neighboring sentences. The relevance of the neighboring sentences again depends partially on their neighbors. Using these two assumptions the recursive ranking function for sentence retrieval was defined as:

$$R_{con}(s|q) = (1-\mu) \cdot R(s|q) + \mu \cdot \left[R_{con}(s_{prev}(s)|q) + R_{con}(s_{next}(s)|q)\right] \quad (2)$$

where $s_{prev}(s)$ depicts previous sentence of sentence s and $s_{next}(s)$ next sentence of sentence s. $R_{con}(s_{prev}(s)|q)$ and $R_{con}(s_{next}(s)|q)$ represent the relevance of the previous and next sentence. $R_{con}(s_{prev}(s)|q)$ is by definition 0 if s is first sentence in document and $R_{con}(s_{next}(s)|q)$ is by definition 0 if s is last sentence in document. μ is a tuning parameter. In our tests in this paper the recursive function calls itself until three previous and three next sentences of the sentence s are involved. In other words three recurrences are used. After that no context is used i.e. $R_{con}(s|q) = R(s|q)$ and the recurrence stops. In [18] this sentence retrieval method was called TF-ISF$_{con}$.

In this paper we hypothesize that TF-ISFcon method will show good performance even when calculated at the document level. More precisely, the standard TF-ISF ranking function ($R(s|q)$) can be seen as a function of the sentence properties and document collection properties. The TF-ISF$_{con}$ranking function ($R_{con}(s|q)$) can be seen as a function of sentence properties, neighbor sentence properties and document collection properties. Both ranking functions have in common that they depend on the whole document collection. So if you want to use them, you need the whole collection. We find it interesting to test if we can achieve competitive performance by limiting the data source used for the ranking function to be the document the sentence came from. The reason for developing such a new method and the application of it is presented in Section 3.
If we better analyze the ranking functions $R(s|q)$ and $R_{con}(s|q)$ we can easily define parts that depend on the whole collection. Those are:

- sf_t or number of sentences that contain term t (in the collection)
- n or number of sentences in the collection

We are now replacing sf_t with sf_t^{DL} and n with n_{DL} where

- sf_t^{DL} is the number of sentences that contain term t in the the document that contains the sentence s
- n_{DL} is the number of sentences in the document that contains the sentence s

At first we define the Document Level TF-ISF (abbreviated as DL TF-ISF) by modifying the baseline TF-ISF method. The corresponding ranking function $R^{DL}(s|q)$ is defined as follows

$$R^{DL}(s|q) = \sum_{t \in q} \log\,(tf_{t,q} + 1) \log\,(tf_{t,s} + 1) \log\left(\frac{n_{DL} + 1}{0.5 + sf_t^{DL}}\right) \quad (3)$$

Secondly we define the Document Level TF-ISF$_{con}$ method (abbreviated as DL TF-ISF$_{con}$) by modifying the TF-ISF$_{con}$. The corresponding ranking function $R_{con}^{DL}(s|q)$ is defined as follows

$$R_{con}^{DL}(s|q) = (1-\mu) \cdot R^{DL}(s|q) + \mu \cdot \left[R_{con}^{DL}\left(s_{prev}(s)\middle|q\right) + R_{con}^{DL}(s_{next}(s)|q)\right] \quad (4)$$

$R_{con}^{DL}\left(s_{prev}(s)\middle|q\right)andR_{con}^{DL}(s_{next}(s)|q)$ represent the relevance of the previous and next sentence. $R_{con}^{DL}\left(s_{prev}(s)\middle|q\right)$is by definition 0 if s is first sentence in document and $R_{con}^{DL}(s_{next}(s)|q)$ is by definition 0 if s is last sentence in document. $\mu$is a tuning parameter. In our tests in this paper the recursive function calls itself until three previous and three next sentences of the sentence s are involved analogous to the ranking function $R_{con}(s|q)$.

2.1 Empirical Study of DL TF-ISF$_{con}$ method

Our aim was to test the performance of the DL TF-ISF$_{con}$in comparison to the baseline TF-ISF, to the DL TF-ISF and to the TF-ISF$_{con}$.

We tested our new method for sentence retrieval using data from the three TREC Novelty tracks provided from 2003 to 2004 [3-5]. The task was novelty detection consisting of two subtasks, finding relevant sentences and finding novel sentences. We are only interested in finding relevant sentences i.e. sentence retrieval. Sentence retrieval is an important part of the novelty detection. Allan [8] showed that the performance of the novelty detection depends on the quality of the performance of sentence retrieval.

In each of the three Novelty Tracks in the years 2002, 2003 and 2004 the task was as follows: given a topic and an ordered list of documents find relevant and novel sentences. Participants got set of 50 topics in each track with each topic consisting of titles, descriptions and narratives. They also got a list of mostly relevant documents and a list of sentence level relevance judgments.
In TREC 2002 the topics from ad hoc Tracks were used. 25 documents were assigned to each topic. If the topic had 25 or more relevant documents, only 25 relevant documents were used. If the topic had less than 25 documents, non-relevant documents were added to reach the number of 25 documents. The participating assessors marked about 2% of the sentences relevant.

In TREC 2003 topics where constructed specially for the Novelty track. 25 relevant documents were chosen for every track. 37.56% of sentences were judged relevant.

In TREC 2004 between 25 and 100 documents were chosen with 25 of them relevant. 16.2% of sentences were judged relevant.

An example of a topic from the TREC 2002 Novelty track is shown in Table 2.

Table 1**.** A topic example from the TREC 2002 Novelty track

Title	International Art Crime
Description	Isolate instances of fraud or embezzlement in the international art trade.
Narrative	A relevant document is any report that identifies an instance of fraud or embezzlement in the international buying or selling of art objects. Objects include paintings, jewelry, sculptures and any other valuable works of art. Specific instances must be identified for a document to be relevant; generalities are not relevant.

In the experiments we used Rapidminer[1] [*19*], an open-source system for data mining with Text Extension[2] that provides the vector space model. With Rapidminer all upper cases were transformed to lower case and standard stop words were removed. Stemming was not applied. Results from Rapidminer were presented as a web service and further used in a custom program that implemented the sentence retrieval methods.

We used short queries from the title field. We measured the performance using three P@X measures (P@10, P@50, P@100) and the standard measures MAP, and R-precision. To compare the difference between the two methods we used two tailed paired t-test with significance level $\alpha = 0.05$.

Our ranking function required tuning of the parameter μ, so we employed a train-test methodology similar to [1]. We experimented with three training-testing configurations using TREC Novelty track data as follows:

- Training with TREC 2002 and testing with TREC 2003 and TREC 2004
- Training with TREC 2003 and testing with TREC 2002 and TREC 2004
- Training with TREC 2004 and testing with TREC 2002 and TREC 2003

Training was performed to find the value of parameter μ for which the system shows best performance. During each of the three trainings (TREC 2002, 2003, 2004) we tried values from $\mu = 0.0$ to $\mu = 1.0$ in steps of 0.05. The best value of μ was fixed in order to apply it to the two remaining data sets. During training we measured the performance of the system by using Mean Average Precision (MAP).

Table 2 shows the optimal parameter values for DL TF-ISF$_{con}$.

Table2**.** Optimal μ values for DL TF-ISF$_{con}$

	$\boldsymbol{\mu}$
TREC 2002	0.05
TREC 2003	0.05
TREC 2004	0.15

The next tables (Table 10, 11 and 12) and graphs (Figure 2, 3, 4) show the results for the three training-testing configurations. In Tables 10, 11 and 12 statistically significant differences in comparison to TF-ISF are marked with an asterisk (*), statistically significant differences in comparison to DL TF-ISF are marked with an † and statistically significant differences in comparison to TF-ISF$_{con}$are marked with an^{C}.

When it comes to the MAP the results are as follows:

- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to TF-ISF in each of the six measurements.
- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to DL TF-ISF in each of the six measurements.
- Statistically significant worse results appeared in one out of six measurements when comparing DL TF-ISF$_{con}$ to TF-ISF$_{con.}$

When it comes to the R-precision the results are as follows:

[1]http://rapid-i.com/content/view/181/196/
[2]http://rapid-i.com/content/view/202/206/

- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to baseline TF-ISF in four out of six measurements with no statistically significant worse results.
- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to DL TF-ISF in each of the six measurements.
- Statistically significant worse results appeared in one out of six measurements when comparing DL TF-ISF$_{con}$ to TF-ISF$_{con.}$

When it comes to the P@X measures (P@10, P@50, P@100) the results are as follows:

- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to baseline TF-ISF in two out of 18 measurements with no statistically significant worse results.
- There is statistically significant improvement when using DL TF-ISF$_{con}$ in comparison to DL TF-ISF in two out of 18 measurements with no statistically significant worse results.
- Statistically significant worse results appeared in two out of 18 measurements when comparing DL TF-ISF$_{con}$ to TF-ISF$_{con.}$

Table 3. P@x, MAP and R-Precision for TREC 2003 and 2004, $\mu = 0.2$ for TF-ISF$_{con}$, $\mu = 0.05$ for DL TF-ISF$_{con}$

	TREC 2003				**TREC 2004**			
	TF-ISF	**DL TF-ISF**	**TF-ISF$_{con}$**	**DL TF-ISF$_{con}$**	**TF-ISF**	**DL TF-ISF**	**TF-ISF$_{con}$**	**DL TF-ISF$_{con}$**
P@10	0.6980	0.6980	0.6980	0.6940	0.4220	0.4360	0.4460	0.4340
P@50	0.6436	0.6452	0.6556	0.6432	0.4040	0.4012	0.4028	0.4008
P@100	0.6078	0.6048	0.6184*†	0.6034[C]	0.3660	0.3572	0.3716†	0.3602
MAP	0.5764	0.5724	0.5930*†	0.5857*†	0.3252	0.3225	0.3398*†	0.3340*†
R-Prec.	0.5457	0.5496	0.5725*†	0.5625*†	0.3376	0.3265	0.3456†	0.3321†

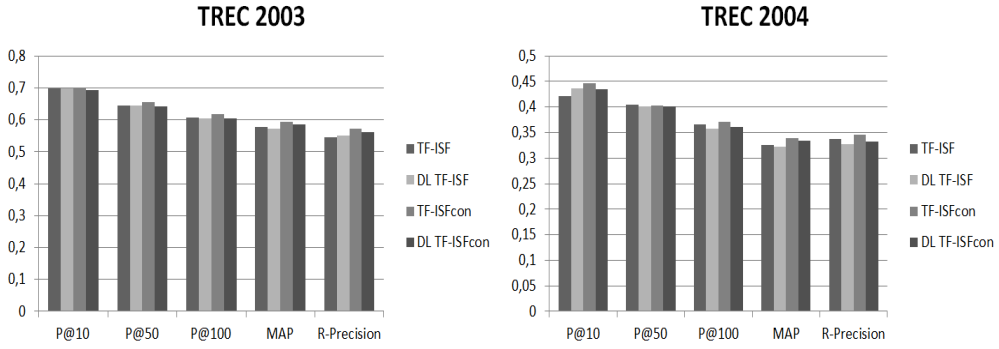

Fig. 2.P@x, MAP and R-Precision for TREC 2003 and 2004, $\mu = 0.2$ for TF-ISF$_{con}$, $\mu = 0.05$ for DL TF-SF$_{con}$

Table 4. P@x, MAP and R-Precision for TREC 2002 and 2004, $\mu = 0.1$ for TF-ISF$_{con}$, $\mu = 0.05$ for DL TF-ISF$_{con}$

	TREC 2002				TREC 2004			
	TF-ISF	**DL TF-ISF**	**TF-ISF$_{con}$**	**DF TF-ISF$_{con}$**	**TF-ISF**	**DL TF-ISF**	**TF-ISF$_{con}$**	**DL TF-ISF$_{con}$**
P@10	0.2900	0.3200	0.3020	0.3280	0.4220	0.4360	0.4340	0.4340
P@50	0.2416	0.2504	0.2488	0.2600	0.4040	0.4012	0.3988	0.4008
P@100	0.1904	0.1914	0.2146*†	0.2134*†	0.3660	0.3572	0.3714†	0.3602
MAP	0.1952	0.2065	0.2315*†	0.2399*†	0.3252	0.3225	0.3392*†	0.3340*†
R-Prec.	0.2414	0.2470	0.2666*†	0.2677*†	0.3376	0.3265	0.3473*†	0.3321*†^{C}

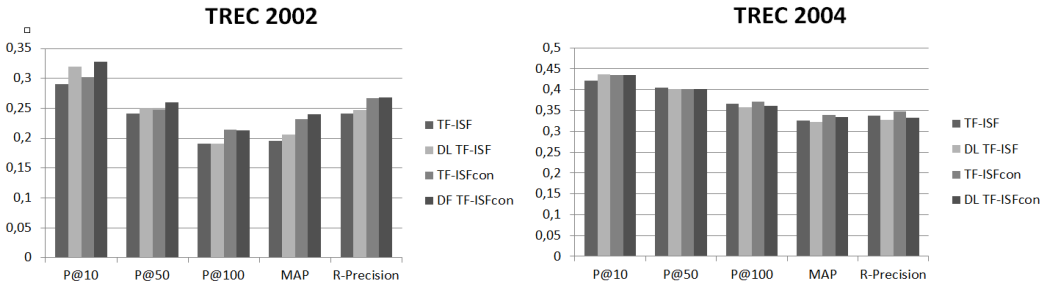

Fig. 3.P@x, MAP and R-Precision for TREC 2002 and 2004, $\mu = 0.1$ for TF-ISF$_{con}$, $\mu = 0.05$ for DL TF-ISF$_{con}$

Table 5. P@x, MAP and R-Precision for TREC 2002 and 2003, $\mu = 0.2$ for TF-ISF$_{con}$, μ=0.15 for DL TF-ISF$_{con}$

	TREC 2002				TREC 2003			
	TF-ISF	**DL TF-ISF**	**TF-ISF$_{con}$**	**DL TF-ISF$_{con}$**	**TF-ISF**	**DL TF-ISF**	**TF-ISF$_{con}$**	**DL TF-ISF$_{con}$**
P@10	0.2900	0.3200	0.3040	0.3260	0.6980	0.6980	0.6980	0.6960
P@50	0.2416	0.2504	0.2496	0.2596	0.6436	0.6452	0.6556	0.6360
P@100	0.1904	0.1914	0.2154*†	0.2132*†	0.6078	0.6048	0.6184*†	0.6062^{C}
MAP	0.1952	0.2065	0.2322*†	0.2399*†	0.5764	0.5724	0.5930*†	0.5839*†^{C}
R-Prec.	0.2414	0.2470	0.2672*†	0.2675*†	0.5457	0.5496	0.5725*†	0.5617*†

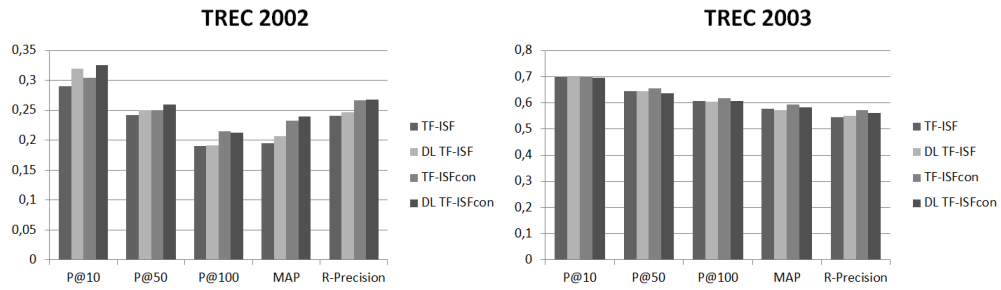

Fig. 4.P@x, MAP and R-Precision for TREC 2002 and 2003, $\mu = 0.2$ for TF-ISF$_{con}$, μ=0.15 for DL TF-ISF$_{con}$

To achieve a better insight into the performance of the DL TF-ISF$_{con}$ in comparison to TF-ISF and to DL TF-ISF we put the data of all three TRECs together. This time we don't have a training data set to pick the best value for μ. For that reason we report the results for a whole range of

values. The results are shown in Table 13 and as graph in Figure 6. In Table 13 statistically significant differences between DL TF-ISF$_{con}$and TF-ISF are marked with an asterisk and statistically significant differences between DL TF-ISF$_{con}$ and DL TF-ISF are marked with a †. This time we can see improvements for a whole range of μ values when using DL TF-ISF$_{con}$ in comparison to TF-ISF and DL TF-ISF ($\mu = 0.1 - 0.3$) when it comes to the MAP and R-Precision. At the same time we do not have statistically significant differences according the P@x values. When choosing higher values of μ ($\mu = 0.4$) we start to get statistically significant worse results according to the P@50 measure and stop getting statistically significant better results according to several MAP and R-Precision measures. This scenario was expected because there must be a threshold value of μ at which the influence of the neighboring sentences is too high.

Table 6. P@X, MAP and R-Precision for the combined data sets of TREC 2002, TREC 2003 and TREC 2004

	TF-ISF	**DL TF-ISF**	**DL TF-ISF$_{con}$**			
			$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$
P@10	0.4700	0.4847	0.4867	0.4907	0.4880	0.4753
P@50	0.4297	0.4323	0.4353	0.4315	0.4287	0.4137*†
P@100	0.3881	0.3845	0.3940†	0.3943†	0.3917†	0.3823
MAP	0.3656	0.3671	0.3867*†	0.3857*†	0.3824*†	0.3732
R-Prec.	0.3749	0.3744	0.3860*†	0.3878*†	0.3841*†	0.3756

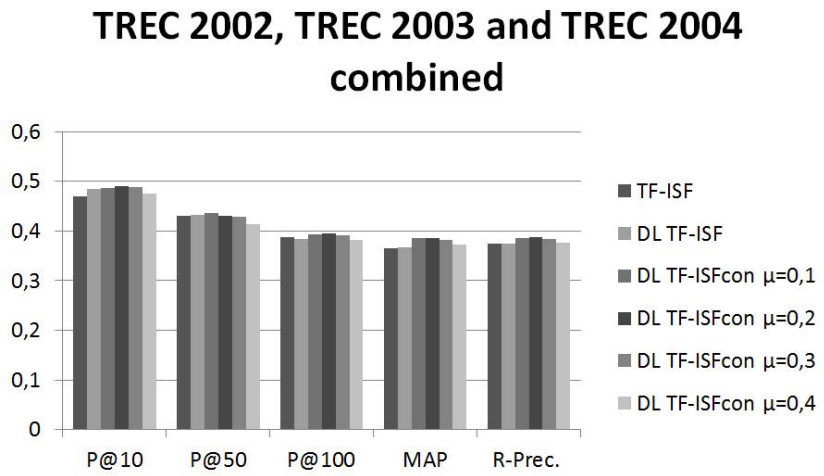

Fig. 5. P@X, MAP and R-Precision for the combined data sets of TREC 2002, TREC 2003 and TREC 2004

We can see from the Table 13. that it is quite easy to find a value of μ that improves the baselines (TF-ISF and DL TF-ISF) according to MAP and R-Precision and at the same time gives competitive results according to reported P@x measures.

The tests in this Section provide evidence for following conclusions.

- The TF-ISF method can be improved using local context according to MAP and R-Precision even when both of them are calculated at document level (see comparison between DL TF-ISF and DL TF-ISF$_{con}$)
- The TF-ISF method can be improved using local context according to MAP and R-Precision even when the baseline TF-ISF is calculated at collection level and the TF-

ISF_{con} is calculated at document level. The baseline TF-ISF uses the whole collection to calculate sentence relevance. Unlike that, the DL TF-ISF_{con} uses only the document that contains the sentence for the same job. Even with that restriction, DL TF-ISF_{con}shows better performance according to MAP and R-Precision.

- There is also some evidence that some useful information is lost when using document level sentence retrieval methods because the method DL TF-ISF_{con} shows sometimes worse results in comparison to TF-ISF_{con} (Table 10 – 12). Despite of that the new method DL TF-ISF_{con} shows better results than the state of the art baseline. In other words the benefit form using context is greater than the drawback from using only document level information.

3. AN OWL DOCUMENT REPRESENTATION FOR SENTENCE RETRIEVAL

We already stated in the Introduction that the aim of semantic web is the usage of standard technologies like OWL to add metadata to the web allowing easier processing of the data on the web and giving the user a better user experience. Key unresolved problems of that vision are the automatic creation of ontologies (Onotology learning) and the automatic instantiation of ontologies (Ontology population). In this chapter we address the problem of automatic population of ontologies concentrating on a specific task (development of a sentence retrieval system). We first manually create an OWL document ontology and then show how to populate it automatically using our previously developed method DL TF-ISF_{con}. Two questions may arise:

1. What is that good for?
2. Why do we need the new method DL TF-$ISF_{con}$and why could we not use just TF-ISFcon or TF-ISF or any other sentence retrieval method that needs entire document collection?

The answer to the first question is: It simplifies development of sentence retrieval systems. Let us imagine an advanced user that wants to develop a sentence retrieval system for example as a browser plugin. Let us also imagine that the user is not interested in every single aspect of information retrieval but at the other side is familiar with the semantic web. To such a user the OWL document representation for sentence retrieval is useful.

The answer to the second question is: Using DL TF-ISF_{con} we can create the OWL document representation for every single document without knowing the other documents. Without it a permanent OWL document representation would not be possible.

Using findings from previous work [18] and Section 2. we suggest that it is possible to develop a semantic web approach to sentence retrieval. In other words it is possible to define an OWL representation of a textual document that can be used for sentence retrieval. How can that been accomplished? Let's look back:

- In [18] we showed how the simple structure of a plain text document (sentence and neighboring sentences) can additionally be exploited to improve the vector space model of sentence retrieval.
- In Section 2 we saw that that it is possible to improve the vector space baseline sentence retrieval method even when the new method uses only data from the document the sentence came from.

Additionally, we have to take into account the characteristics of the vector space model. In a vector space information retrieval model both the document and the query are represented as vectors. A formal representation of the document D and query Q vectors can be defined as follows [20]:

$$d = \left(t_1, w(d,t_1); t_2, w(d,t_2); \dots t_k, w(d,t_k); \dots; t_n, w(d,t_n)\right) \quad (5)$$

$$q = \left(t_1, w(q,t_1); t_2, w(q,t_2); \dots t_k, w(q,t_k); \dots; t_n, w(q,t_n)\right) \quad (6)$$

Where

- n represents number of terms allowed in system,
- $t_1, t_2, \dots t_n$ represents a list of all terms allowed in system,
- $w(d,t_k)$ represents the weight of term t_k in a document d,
- $w(q,t_k)$represents the weight of term t_k in a query q.

Given the vector representations of the document and the query, a similarity value may be obtained using following ranking function:

$$R(d|q) = \sum_{k=1}^{n} w(d,t_k) \cdot w(q,t_k) \quad (7)$$

The term weights $w(d,t_k)$ and $w(q,t_k)$ are defined using a term frequency component, inverted document frequency component and a normalization component [21]. One example of such a ranking function applied to sentence retrieval is TF-ISF (equation 1).

When it comes to the OWL representation of a document for the sentence retrieval task we are interested in expressing the importance of a term to a sentence. If we focus on the document vector d we can see that the importance of a term t to a document d can be expressed using the following natural language statement:

"Document d contains term t with weight $w(d,t)$."

Analogously the importance of a term t to a sentence s can be expressed using the following statement:

"Sentence s contains term t with weight $w(s,t)$".

If we apply this logic to the DL TF-ISF_{con}method then two kinds of statements are possible depending on whether the term appears in the sentence or the context:

1. If the term appears in the sentence
   - "Sentence s contains term t with weight $w(s,t)$".
2. If the term appears in the context (neighbor sentences)
   - "Sentence s contains in context term t with weight $w_{con}(s,t)$."

One example of such sentence would be "Sentence <*s docid="NYT19981017.0086" num="4"*> contains term *spain*with weight *0,00725*."

When it comes to the weight $w(s,t)$ we define it to be the following TF and the ISF component taken from equation 3.

$$w(s,t) = \log\left(tf_{t,s} + 1\right) \log\left(\frac{n_{DL} + 1}{0.5 + sf_t^{DL}}\right) \quad (8)$$

When it comes to the weight $w_{con}(s,t)$ we define it to be the TF and ISF components related to the previous and next sentence taken from relation 4 as follows.

$$w_{con}(s,t) = w_{rec}(s_{prev},t) + w_{rec}(s_{next},t) \quad (9)$$

Where $w_{rec}(s_{prev},t)$ and $w_{rec}(s_{next},t)$ are defined as follows

$$w_{rec}(s_{prev},t) = (1-\mu)\cdot w(s_{prev},t) + \mu\cdot\left[w_{rec}\left(s_{prev_{prev}},t\right) + w_{rec}\left(s_{prev_{next}},t\right)\right] \quad (10)$$

$$w_{rec}(s_{next},t) = (1-\mu)\cdot w(s_{next},t) + \mu\cdot\left[w_{rec}\left(s_{next_{prev}},t\right) + w_{rec}\left(s_{next_{next}},t\right)\right] \quad (11)$$

Where $w_{rec}(s_{prev},t)$ is by definition 0 if s is first sentence in document and $w_{rec}(s_{next},t)$ is by definition 0 if s is last sentence in document. $s_{prev_{prev}}$depicts previous sentence of the s_{prev}.$s_{prev_{next}}$depicts next sentence of sentence s_{prev}. $s_{next_{prev}}$depicts previous sentence of sentence s_{next}. $s_{next_{next}}$depicts next sentence of the sentence s_{next}. The base case for which the functions $w_{rec}(s_{prev},t)$ and $w_{rec}(s_{next},t)$ produce result without recurring is omitted for readability. We define it the same as in Section 2.by taking into account the number of times the function called itself. When the recurring is reached that includes three previous and three next sentences of the sentence s the recurring stops (e.g. $w_{rec}\left(s_{prev_{prev_{prev}}},t\right) = w\left(s_{prev_{prev_{prev}}},t\right)$). In our tests in Section 2. we did not try to include a higher number of neighboring sentences into the computation of the relevance of a sentence. Of course it is possible to take into account all the neighboring sentences in a document. Determining the optimal number of previous and following sentences is left for future work.

Now we can start coding our two sentences ("Sentence s contains term t with weight $w(s,t)$." and "Sentence s contains in context term t with weight $w_{con}(s,t)$.") using OWL. A ternary relation that connects the sentence, the term and the weight has to be used. To define a ternary relation we use a method for *representing additional attributes describing a relation* presented by Noy et al. in [22]. To realize the ternary relation we define two classes:

- Sentence
- TermImportance

All sentences from the document are members (instances) of the class Sentence. Every sentence is connected to a string (type rdfs:Literal) that contains its plain text content through the relation:

- hasContent

Every sentence instance is connected to instances of the class TermImportance through the following relations:

- contains
- containsInContext

To make a ternary relation complete instances of TermImportance are connected to the term name (type rdfs:Literal) and to the weight (type xsd:double) through the following relations

- hasTermName
- hasWeight

Figure 6. shows classes, data types and properties used to represent a sentence for the task of sentence retrieval.

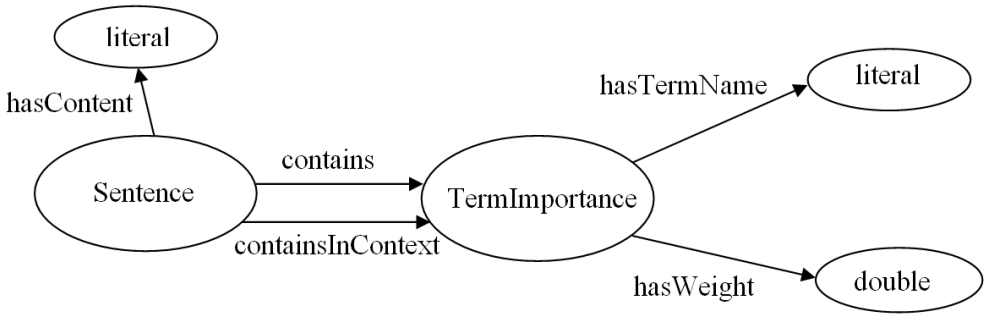

Fig. 6. Representation of a sentence used for document retrieval with classes, data types and properties.

It is straightforward to automatically create instances of the above classes and automatically create property assertions (see Figure 6.). Every sentence instance has multiple *contains* and *containsInContext* assertions depending on whether a term appears in the sentence or its neighboring sentences. The string (type rdfs:Literal) of the *hasTermName* assertion is just the term name after some preprocessing (e.g. with all letters converted to a lowercase). The real number (type double) of the *hasWeight* property assertion is calculated using equation 8 if we are talking about a term from the sentence and using equation 9 if we are talking about a term from the context. A textual document is represented using multiple sentences where each sentence is connected to multiple term names and weights. If a term does not appear in a sentence or in the context, a *contains* or *containsInContext* assertion is omitted respectively. Using such a document representation the ranking function can been defined as

$$R_{con}^{DL}(s|q) = (1-\mu)\cdot\sum_{t\in q}\log\left(tf_{t,q}+1\right)\cdot \mathrm{w(s,t)} + \mu\cdot\sum_{t\in q}\log\left(tf_{t,q}+1\right)\cdot w_{con}(s,t) \qquad (12)$$

The benefits of such a document representation for sentence retrieval are as follows

- It simplifies development of sentence retrieval allowing the user to concentrate only on standard OWL reading and simple math. The simplification comes from the fact that partial results of the sentence retrieval process are stored for future use. More precisely subtasks like preprocessing (stop word removal, converting to lowercase, stemming etc.) or TF-ISF calculation are explicitly recorded for future use using the standard OWL. In other words the user that wants to develop a sentence retrieval system is freed from preprocessing and TF-ISF calculation.

- We believe that our OWL document representation has the potential to a wider application. The OWL document representation could be useful to anybody interested in TF-ISF values. The significance of the TF-IDF (which is similar to TF-ISF) value goes beyond the borders of information retrieval. For example in [23] words with high TF-IDF values were used as most discriminative keywords for association rule mining from text. We believe that our document representation can also be used to simplify implementation of association rule mining from text. Details of such an implementation are left for future work.

- We saw that we are able to automatically populate the presented document ontology which is a new way of ontology population that is based on sentence retrieval and it presents an additional contribution in this paper.

4. CONCLUSIONS

In this paper we restricted our previous TF-ISF$_{con}$ method to use only document level information and called it DL TF-ISF$_{con}$. This method like TF-ISF$_{con}$ also showed better performance according to the MAP and R-Precision than the TF-ISF baseline. The new document level method allows calculating the relevance score of a sentence using only the document that contains the sentence. That characteristic allowed us to propose an OWL document representation for sentence retrieval that can be considered as sentence retrieval document ontology. It was shown how it is possible to fully automatically populate the ontology with document level data. The benefits of the OWL document representation lie down in the possibility to permanently save partially results of the sentence retrieval process like preprocessing and TF-ISF values. All that simplifies development of future sentence retrieval systems and other systems that use TF-ISF values. Additional contribution is the new way of automatic ontology population based on sentence retrieval.

REFERENCES

1. Fernandez RT, Losada DE, Azzopardi LA (2010) Extending the language modeling framework for sentence retrieval to include local context, Information Retrieval, 14(4), pp. 355-389
2. Murdock VG (2006). Aspects of sentence retrieval. PhD thesis, University of Massachusetts Amherst.
3. Harman D (2002) Overview of the TREC 2002 novelty track. In: Proceedings of the Eleventh Text Retrieval Conference (TREC)
4. Soboroff I, Harman D (2003) Overview of the TREC 2003 novelty track. In Proceedings of the Twelfth Text Retrieval Conference (TREC)
5. Soboroff I (2004) Overview of the TREC 2004 novelty track. In Proceedings of the Thirteenth Text Retrieval Conference (TREC)
6. Voorhees EM (2003) Overview of the TREC 2003 Question Answering Track. TREC 2003: 54-68
7. Daumé H, Marcu D (2006) Bayesian query-focused summarization. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp. 305-312, Sydney, Australia.
8. Allan J, Wade C, Bolivar A (2003) Retrieval and novelty detection at the sentence level. In Proceedings of the 26th ACM international conference on research and development in information retrieval (SIGIR 2003) (pp. 314–321)
9. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Communications of the ACM, 18, 613–620.
10. Losada DE, Fernandez RT (2007) Highly frequent terms and sentence retrieval. In Proceedings of the 14th String processing and information retrieval symposium (SPIRE 2007), Lecture Notes in Computer Science (pp. 217–228). Santiago de Chile, Chile: Springer.
11. Fernandez RT, Losada DE (2009) Using opinion-based features to boost sentence retrieval. In Proceedings of the ACM 18th conference on information and knowledge management (CIKM 2009) (pp. 1617–1620). Hong Kong, China: ACM.
12. Ponte J, Croft WB (1998) A language modeling approach to information retrieval. In Proceedings of the 21st Annual Conference on Research and Development in Information Retrieval (ACM SIGIR)
13. Tsai M-F, Chen H-H (2002) Some Similarity Computation Methods in Novelty Detection. In Proceedings of the Eleventh Text Retrieval Conference. Gaithersburg, NIST Special Publication, pp. 500-251
14. Losada DE (2008) A study of statistical query expansion strategies for sentence retrieval. In Proceedings SIGIR 2008 workshop on focused retrieval (question answering, passage retrieval, element retrieval), Singapore: ACM.

15. Collins-Thompson K, Ogilvie P, Zhang Y, Callan J (2002) Information filtering, novelty detection and name-page finding. In Proceedings of the 11th text retrieval conference (TREC 2002), Gaithersburg, Maryland, November 2002.
16. Berners-Lee T, (1999) Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor, Harper San Francisco.
17. Gruber T (1993) A translation approach to portable ontology specification. Knowledge Acquisition 5(2), 199–220.
18. Doko A, Stula M, SericLj (2013) Improved sentence retrieval using local context and sentence length. Information Processing & Management 49(6). (pp.1301-1312)
19. Mierswa I, Wurst M, Klinkenberg R, Scholz M, Euler T (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks, In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.
20. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 5
21. Manning CD, Raghavan P, Sch utze H (2008) Introduction to Information Retrieval.Cambridge University Press, Cambridge, UK.
22. Noy N, Rector A, Hayes P, Chris W (2006) Defining N-ary Relations on the Semantic Web. W3C Working Group Note 12 April 2006.
23. Mahgoub H, Roesner D, Ismail N, Torkey F (2008) A Text Mining Technique Using Association Rules Extraction. International Journal of Information and Mathematical Sciences 4:1

AUTHORS

Alen Doko was born 19.06.1982 in Mostar, Bosnia and Herzegovina. He received his Graduate Engineer Degree in 2007 and a PhD in 2013. He is also a Business and Analytical Information SystemsAssociate at JP Croatian Telecommunication d.o.o. Mostar, Mostar, Bosnia and Herzegovina. His research interests include Semantic Web, Information Retrieval and Sentence Retrieval. He is author and coauthor of four scientific papers.

Maja Štula is full professor of computer science at the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. She received a BS in electrical engineering in 1996, a MSc in 2001 and a PhD in 2005. all from the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. Her research interests include intelligent technologies like multi-agent systems, ontology and fuzzy cognitive maps, especially engineering applications of intelligent technologies. Contact information: FESB, R. Boskovica 32, 21000 Split, Croatia; maja.stula@fesb.hr

LjiljanaŠerićis a professor of computer science at the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. She received aa PhD in 2010. She is author and coauthor of many scientific papers (http://bib.irb.hr/lista-radova?autor=272906).