Snippet Generation for Semantic Web Search Engines

Thomas Penin[1], Haofen Wang[1], Thanh Tran[2], and Yong Yu[1]

[1] Department of Computer Science & Engineering
Shanghai Jiao Tong University, Shanghai, 200240, China
{tpenin,whfcarter,yyu}@apex.sjtu.edu.cn
[2] Institute AIFB, Universität Karlsruhe, Germany
{dtr}@aifb.uni-karlsruhe.de

Abstract. With the development of the Semantic Web, more and more ontologies are available for exploitation by semantic search engines. However, while semantic search engines support the retrieval of candidate ontologies, the final selection of the most appropriate ontology is still difficult for the end users. In this paper, we extend existing work on ontology summarization to support the presentation of ontology snippets. The proposed solution leverages a new semantic similarity measure to generate snippets that are based on the given query. Experimental results have shown the potential of our solution in this problem domain that is largely unexplored so far.

Keywords. Snippet, Ontology summarization, Semantic measure

1 Introduction

More and more ontologies are available on the Semantic Web. A significantly growing part is concerned with specific domains comprising ontologies designed to be useful to companies. To develop a semantic application, an engineer can draw from a large set of reusable ontologies. However, a main question remains: how to find and select the exact ontology matching the requirements? While the retrieval of potential candidate ontologies can be conveniently achieved through semantic search engines such as Sindice[3], Falcons[4], Swoogle[5], Watson[6], etc., the final selection still presents to be a difficult problem.

Our engineer can be considered as a *content curator* [1]. While he may be an expert in his domain, he does not necessarily have a deep understanding of Semantic Web technologies. When considering the result page of a search engine, his concern is to find out how these documents representing ontologies entail the query, which topics are covered, and if there are classes missing or that should not be included in the solution.

[3] http://www.sindice.com/
[4] http://iws.seu.edu.cn/services/falcons/objectsearch/index.jsp
[5] http://swoogle.umbc.edu/
[6] http://watson.kmi.open.ac.uk/WatsonWUI/

To avoid the burden of downloading all potentially interesting documents and to be confronted to the even more laborious process of opening them with an ontology visualization tool to examine their internal organization, most of the search engines offer certain facilities to get an idea of the ontology content from the result page. Sindice provides for example the main topic and metadata that can be explored by the user. Falcons associates labels with every document, while Swoogle displays an extract of the classes' names related to the searched terms. Watson offers a list of instances and classes matching the query.

Despite these features, these systems do not seem to exactly match the needs of the engineer. Sindice is limited to one topic per ontology and often only presents the document title. While the extracted classes' names in Swoogle can provide useful information, they can not serve as an adequate overview of the given ontology. Labels provided by Falcons are simply derived from the file name while Watson does not consider other resources than those related to the query.

To address the needs of our engineer, we propose a new snippet generation system for semantic web documents (ontologies), that brings the following contributions:

- A new measure for assessing the similarity of RDF sentences which is exploited for topic identification. Contrary to most of the measures we are aware of, this measure does not limit its scope to nouns from entities and class names but includes verbs and adjectives from triple's subjects, predicates, and objects. It does also not only consider sentences as bags of words, but use their internal structure to improve its accuracy.
- An extension of the work of [2] on ontology summarization and [3] on semantic similarity between words, showing that both can be successfully applied to the problem of snippet generation.
- A snippet generation system that can be tested through the web interface at `http://snippet.apexlab.org`.
- A user evaluation shows that our approach to snippet generation brings about promising results.

This paper will be organized as follows. In section 2, we will describe the elements and structures that will be used to define our semantic similarity measure in section 3. Section 4 will then describe the snippet generation process and present the structure of our system and its test interface. The quality and the efficiency of our solution will be explored in section 5. Finally, section 6 will discuss the related research and section 7 will conclude about the future work.

2 RDF Sentence Graph and Topic Graph

In this section, we present the definitions of RDF sentences and RDF sentence graphs, as discussed in [2]. Then, we extend this work to define RDF topics and RDF topic graphs.

Let O be an ontology, we call T the set of its RDF triples and B the set of its blank nodes. For $t \in T$, we note by $subj(t)$, $pred(t)$ and $obj(t)$ the subject,

the predicate and the object of t respectively. We define the set of triples of O that contains blank nodes by $T_B = \{t \in T,\ subj(t) \in B \text{ or } obj(t) \in B\}$. It is clear that $T_B \subseteq T$. For $b \in B$, let $T_b \subseteq T_B$ be the subset of triples containing b.

We say that $t_i, t_j \in T$ are b-connected if they satisfy one of the following conditions:

- $\exists\ b \in B$ such that $t_i,\ t_j \in T_b$;
- $\exists\ t_k \in T$, such that t_i and t_k are b-connected and t_j and t_k are b-connected.

Definition 1. *An RDF sentence s is a set of triples such that:*

1. *$\forall i, j \in s$, i and j are b-connected;*
2. *$\forall i \in s$, $\forall j \notin s$, i and j are not b-connected.*

Intuitively, a RDF sentence is formed by a main RDF statement and all the other RDF statements b-connected to it. In particular, a RDF statement whose subject is not a blank node is called a main RDF statement. We call S the set of RDF sentences of O. For all $s \in S$, we define:

- $Subj(s) = \{subj(t)$ such that $t \in s$ and $subj(t) \notin B\}$;
- $Pred(s) = \{pred(t)$ such that $t \in s\}$;
- $Obj(s) = \{obj(t)$ such that $t \in s$ and $obj(t) \notin B\}$.

A reason to choose sentences rather than triples is to avoid the case of subjects and objects that are blank nodes. This would have made it more difficult to paraphrase them with natural language.

RDF sentences preserve the connections of the initial RDF graph. In order to rank RDF sentences, we can define links between them, based on their common nodes. [2] proposed to consider two kind of links (*sequential* – predicate or object of a sentences being the subject of another – and *coordinate* – several sentences with the same subject) and defined a parameter p to assign importance to each of them.

We reuse these notions of links to obtain what is called an *RDF sentence graph*. Like [2], we define the weight of the link from $s_1 \in S$ to $s_2 \in S$ by

$$w(s_1, s_2) = p * seq(s_1, s_2) + (1 - p) * cor(s_1, s_2), \tag{1}$$

where $seq(s_1, s_2)$ and $cor(s_1, s_2)$ are equal to 0 or 1, respectively depending on the existence or not of a sequential and a coordinate link between both sentences.

In an ontology, it is often possible to identify several topics. In a text document, a topic can be defined as a set of words or sentences sharing a certain semantic proximity. Similarly, we extend [2] and come with the following definition:

Definition 2. *Given a threshold θ, an RDF topic is defined as a set of RDF sentences such that the pairwise semantic similarity between these sentences is greater than or equal to θ.*

A definition of the similarity considered here is given in section 3.

As for sentences, we build an *RDF topic graph*. Let T_1 and T_2 be two topics and $|T_i|$ the number of sentences of T_i. Using equation 1, we define the weight of the link between T_1 and T_2 by

$$w(T_1, T_2) = \frac{1}{|T_1| + |T_2|} * \sum_{(s_i, s_j) \in T_1 \times T_2} w(s_i, s_j). \quad (2)$$

The main idea behind this definition is to be able to transfer the weight of the links between sentences to their topics. Note that we make use of an average weight.

3 Semantic Similarity Measure

Our main goal is to derive the aboutness of an ontology, i.e. the topics covered by an ontology. To do so, there is the need for a semantic similarity measure that can be used to decide whether two given RDF sentences belong to the same topic or not.

There already exist a lot of measures computing the similarity between ontological structures in order to achieve tasks such as ontology selection or alignment. Two commonly taken approaches are a comparison at the conceptual level (ontological structure) or at the lexical level (vocabulary). In some work, e.g. [4], both these levels are leveraged. Among the drawbacks of comparing ontologies at the conceptual level, there is the fact that structures depend on the way the ontologies are built. Patterns can differ and do not always reflect the complete meaning since concepts are usually not only expressed by logical structures but also by the choice of the vocabulary. Moreover, it is often required to consider structures complex enough to draw meaningful conclusions. To improve the relevance, works like [4] introduce lexical level comparison, using string [5] or semantic [6, 3, 7] similarity metrics. Such propositions rely however on taxonomies and often only compare class and instance names, and consider them as simple bags of words.

We propose another solution gathering both approaches, based on the definition of RDF sentence given in section 2. This choice allows a comparison between structures that have the same pattern whatever the ontology and whatever its size through the definition of subjects, predicates and objects for RDF sentences. Conceptualization is embedded within each sentence and concepts are not limited to class and instance names. Instead, the whole content of the ontology is available. Moreover, as explained in the current section, sentences are not limited to bags of words and their internal semantics is used to measure their similarity. Rather than relying on a basic string similarity, we use the semantic similarity defined between two words by [3]. Our choice was based on the performance of their solution compared with other propositions like [6] and [7]. However, we slightly extended its functionalities to be able not only to compare nouns, but also verbs, adjectives and adverbs, which is something commonly ignored by the state-of-the-art.

To design our measure, we try to apply some principles proposed by [8]: consider both commonalities and differences between sentences and ensure that the maximum is reached when they are identical. [5] gave other interesting features. A measure shall be fast (polynomial), stable, intelligent and discriminating. Section 5 shows how much we achieved these goals.

3.1 Similarity between two lists of words

To compare sentences, we first need to be able to compute similarity between lists of words, considered as bags of words. Lists of words may or may not share common concepts and can have different length. Our idea, following [8], is to consider both commonalities and differences.

Let w be a given word and $w_1, \cdots, w_n$ be a series of words belonging to a list L. We define the semantic similarity between w and L by

$$sim(w, L) = \max \{sim(w, w_i) \text{ for } i = 1..n\}, \tag{3}$$

where $sim(w, w_i)$ is the result returned by the measure defined by [3].

The word w is related to concepts that may or may not be found in L. By computing the similarity with any word of L, we will get a high score if the list contains a concept close to w. If not, the score will be low, even with the max function. The maximum allows to prove that w and L share commonalities or differences, without actually considering their respective weight.

To improve the efficiency of the measure, we decided not only to consider nouns but also verbs, adjectives and adverbs. We used the WordNet taxonomy to find the noun and the verbal base corresponding to each given adjective, adverb or conjugated verb. This approach consisting to only compare nouns and verbal base can be considered as a certain loss of semantic meaning. However, we consider that it is far better than simply ignore these words and that a significant part of the meaning is still conveyed by the nouns.

Let L_1 and L_2 be two lists of words with respective length n_1 and n_2. Let w_{ij} be the i^{th} element of the list L_j. The semantic similarity between L_1 and L_2 is defined by

$$sim(L_1, L_2) = \frac{1}{n_1 + n_2} \left(\sum_{i=1}^{n_1} sim(w_{i1}, L_2) + \sum_{j=1}^{n_2} sim(w_{j2}, L_1) \right), \tag{4}$$

where sim is defined by equation 3.

Now, if k pairs of words share the same semantic meaning, they will account for k times in the overall semantic similarity, which ensures that the weight of the different concepts is taken into consideration. We can see that if a concept is only present in one list and not in the other, it will tend to reduce the overall result. Finally, the average function keeps our measure between 0 and 1. We can clearly see that if the two lists have nothing in common, their similarity will be 0 and that it will be 1 if and only if they exactly share the same concepts.

3.2 RDF Sentence Semantic Similarity

According to section 2, an RDF sentence s can be seen as a virtual triple constituted of three elements: its subject $Subj(s)$, its predicate $Pred(s)$, and its object $Obj(s)$. Each of these elements is a list of words. An immediate benefit is that it is possible to consider RDF sentences for what they really are: sentences that have a meaning, and that similarly to natural language sentences have a subject and an object linked by a predicate.

In natural languages however, subject, predicate, and object do not share the same importance. This seems to offer interesting perspectives if we consider the opportunity to modify the different coefficients to give to different possible matches: subject-subject, object-object, predicate-predicate, or subject-object. Subject-object comparison is important to apprehend chiasms. Consider the sentences *Alice hasMother Anne* and *Anne hasDaughter Alice*. Direct subject-subject and object-object matches would lead to the conclusion that they are not semantically close, which is a mistake. Comparing their subjects with their objects shows that they are in fact very similar.

We define w_{ss}, w_{pp}, w_{oo}, and w_{so} as being the respective weights attributed to subject-subject, predicate-predicate, object-object, and subject-object comparisons. To ensure that our measure will stay between 0 and 1, we assume that

$$w_{ss} + w_{pp} + w_{oo} = w_{pp} + 2 * w_{so} = 1. \tag{5}$$

Let s_1, $s_2 \in S$. Let $sim_{ss}(s_1, s_2)$, $sim_{pp}(s_1, s_2)$, $sim_{oo}(s_1, s_2)$, $sim_{s_1o_2}(s_1, s_2)$, and $sim_{s_2o_1}(s_1, s_2)$ be respectively the similarity between $Subj(s_1)$ and $Subj(s_2)$, $Pred(s_1)$ and $Pred(s_2)$, $Obj(s_1)$ and $Obj(s_2)$, $Subj(s_1)$ and $Obj(s_2)$, and $Subj(s_2)$ and $Obj(s_1)$. Since subjects, objects and predicates are lists of words, this similarity is computed like explained in section 3.1.

Definition 3. *To be able to handle chiasms, we consider that if* $sim_{ss}(s_1, s_2) + sim_{oo}(s_1, s_2) \geq sim_{s_1o_2}(s_1, s_2) + sim_{s_2o_1}(s_1, s_2)$*, the semantic similarity between* s_1 *and* s_2 *is defined by*

$$sim(s_1, s_2) = w_{ss} * sim_{ss}(s_1, s_2) + w_{pp} * sim_{pp}(s_1, s_2) + w_{oo} * sim_{oo}(s_1, s_2).$$

Otherwise, it is defined by

$$sim(s_1, s_2) = w_{so} * (sim_{s_1o_2}(s_1, s_2) + sim_{s_2o_1}(s_1, s_2)) + w_{pp} * sim_{pp}(s_1, s_2).$$

In our system, very common words such as RDF keywords are excluded, since they can give a high similarity to sentences that significantly differ in their meaning.

3.3 Topic Similarity

Topics are lists of semantically close RDF sentences. As such, semantic similarity between two topics is obtained like that between two lists of words:

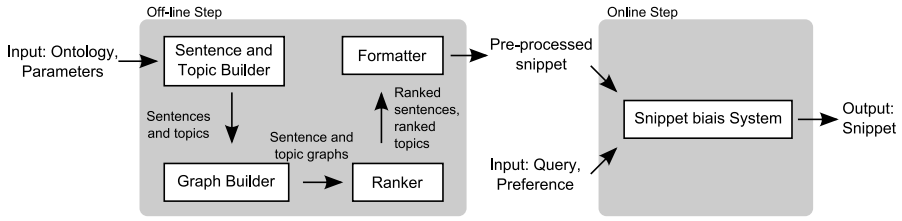


Fig. 1. Architecture of our snippet generation system.

Definition 4. *Let T_1 and T_2 be two topics. Let L_1 and L_2 be the lists of their RDF sentences, with respective length n_1 and n_2. Let s_{ij} be the i^{th} element of the list L_j. The semantic similarity between T_1 and T_2 is defined by*

$$sim(T_1, T_2) = \frac{1}{n_1 + n_2} \left(\sum_{i=1}^{n_1} sim(s_{i1}, L_2) + \sum_{j=1}^{n_2} sim(s_{j2}, L_1) \right)$$

where the similarity between a sentence and a list of sentences is the maximum similarity between this sentence and all the sentences of the list.

4 Snippet Generation Process

As explained in section 1, our system was designed to give the user ways to quickly find out whether an ontology suits his needs or not. It required the determination of the different topics and a ranking of both topics and RDF sentences in a snippet matching user preference. The determination of the different topics is achieved thanks to a hierarchical clustering algorithm. This choice was made since no knowledge of the number of topics is required in advance (the topic threshold is enough) while still being simple to implement. Some performance issues (see section 5) were however raised by experimental results, making it incompatible with the generation of a snippet that should almost instantly displayed once the results found by the search engine are known. Operations needed are more complex than in the case of traditional snippets for text documents [9].

Since most of the work of ontology summarization is unrelated to the query provided by the user, we increased the response time of our interface by splitting the process into two steps: one off-line and one online. The parsing of the ontology file is achieved by the Jena library[7].

The figure 1 shows the general organization of our system.

4.1 Off-Line Step

The *sentence builder* creates RDF sentences from the triples of an ontology. The *topic builder* gathers semantically similar sentences, given a threshold θ. Two

[7] `http://jena.sourceforge.net/`

sentences or two topics with a similarity higher than θ will be merged into the same topic.

The *sentence graph builder* takes then the sentences and build the sentence graph. The *topic graph builder* is in charge of building a topic graph using the sentence graph.

The *sentence ranker* ranks sentences within each topic, using the in-degree centrality in the sentence graph as salience criteria. [2] have shown that the in-degree centrality gave the best ranking results as soon as $p \geq 0.3$ in the sentence graph (see section 2). We take $p \geq 0.7$ and apply their algorithm. Even if a module of our demo system allows the user to change p, this is in practice not necessary. The *topic ranker* ranks the topics considering the topic graph and the in-degree centrality of its nodes.

The *formatter* takes the RDF sentences and apply natural language processing techniques in order to make them more easily readable by the final user. The result is output to the disk for further use. In our demonstration system, pre-processed snippet are stored under XML format and loaded for the online step of the snippet generation process.

4.2 Online Step

The *snippet bias* is applied when the snippet is generated. Sentences matching the user's query are selected. It is made sure that they will be visible in the final snippet, while respecting the user preference. As described in the section 4.4, several options are available to customize both the length of the snippet as well as the relevance of its content.

4.3 Natural Language Output

To improve the readability of the snippet, a system inspired from [10] and [11] was implemented to generate NL sentences. RDF sentences themselves are transformed into triples composed of a subject, a predicate and an object, obtained by considering all possible path in the sentence and by aggregating successive predicates.

RDF keywords are replaced by more natural formulations. `X subClassOf Y` becomes for instance `X is a kind of Y`. Sentences matching certain patterns are also transformed. The patterns are obtained by parsing the sentence using WordNet.

4.4 Snippets

In section 1, we considered some questions that the snippet shall answer.

For the user to identify the topics, they are clearly separated, their respective weight is given and their importance is shown by their rank. Within each topic, sentences are written using NL techniques and ranked according to their salience. The user has the possibility to choose the maximum number of topics

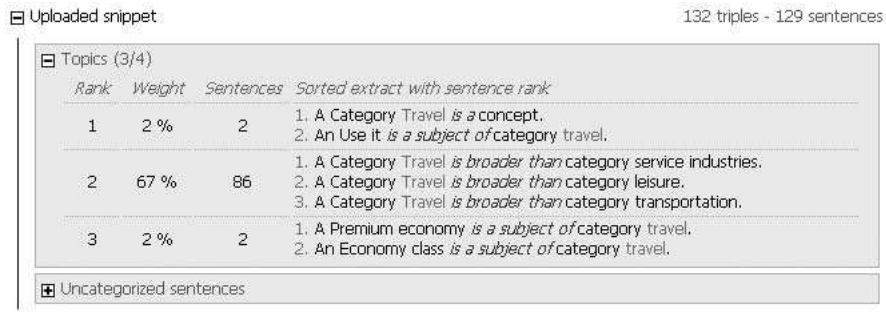


Fig. 2. Sample snippet for `Travel.rdf` for the query *travel*.

and sentences per topic to display. Different degrees of summarization can then be proposed. Finally, only query-related topics and sentences can be displayed or a rule can be applied, ensuring that the most salient topics and sentences will always be shown as soon as at least half of them are related to the query.

Figure 2 shows an example of snippet.

4.5 Test Interface

A demonstration system was implemented[8] as a Java web application. It simulates a search engine result page for predefined queries, allows the upload of an ontology, the personalization of all the parameters of the process to get a query-biased snippet and provides a test component for our semantic similarity measure.

5 Evaluation

To assess the quality of the system, we successively investigated the performance of our similarity measure, the quality of the snippets and to what extend the solution was promising from a user point of view.

The tests were carried out through our demonstration interface (see section 4.5) on our local gigabyte network. The server was a 2.4 GHz personal computer, with 1 GB memory, running Microsoft Windows XP.

Our test panel was composed of nine members from our laboratory. They have a certain familiarity with ontologies without being experts, and as such were more interested in technical details than the end-users targeted by our solution. This choice was made because the technical capacities required by the tests did not match those of casual end-users. Since our system is not yet integrated into a functional semantic search engine, we could not really test according to the use case as defined in section 1. As a result, the bias was not considered as too important.

[8] `http://snippet.apexlab.org`. Tested with Firefox

Ontology	Topic Number	Vocabulary	Structure
AKTiveSAOntology.owl	High	Complex	Complex
animalsA.owl	Low	Simple	Complex
cv.rdfs	Low	Simple	Simple
History_of_China	Low	Complex	Simple
terrorism.owl	High	Complex	Complex
Travel-OilEdExportRDFS.rdfs	High	Simple	Simple

Table 1. Composition and characteristics of the test set

5.1 Semantic Similarity Measure

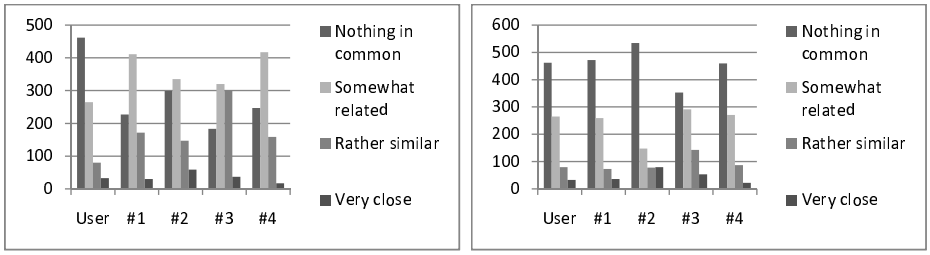
This evaluation aimed both at assessing that our measure can match the appreciation of our test panel and at finding the right parameters to do so.

We selected six ontologies , as shown by table 1. To get close to the diversity of the real Semantic Web, they were chosen to exhibit different characteristics:

- *Topic number.* Since the overall semantic of the ontology is not considered to express the similarity between sentences, this aimed at investigating the stability of the measure and its coherence with the user opinion for different levels of homogeneity. We considered ontologies with more than 30 topics as having a high number of topics.
- *Vocabulary complexity.* It ranges from commonly used words to highly specific terms. This was thought to make sure that our measure stay coherent when considering sentences containing numerous unknown words. We considered vocabulary to be complex when more than 10% of the words were unknown.
- *The internal structure of the ontology.* While some contain complex relations between classes and entities, some offer a catalog-like organization (qualified here as simple). Ontologies from DBpedia for instance contain RDF sentences that often have very similar objects (category name). This should not harm the capacity of the measure to mimic human judgment.

For each ontology, testers were asked to estimate the similarity between pairs of randomly-extracted sentences. They had the choice between *"nothing in common"*, *"somewhat related"*, *"rather similar"* and *"very close"*. Simultaneously, the system computed the similarity for different parameter configurations ♯1, ♯2, ♯3 and ♯4, defined with w_{ss}, w_{pp}, w_{oo} and w_{so} values respectively equals to 0.7, 0.1, 0.2 and 0.45 (strong subject, weak predicate), 0.3, 0.4, 0.3 and 0.3 (strong predicate), 0.45, 0.1, 0.45 and 0.45 (weak predicate) and 0.6, 0.2, 0.2 and 0.4 (strong subject).

Figure 3 (a) describes the opinion of the user and the system judgment for the different configurations. All results between 0.0 and 0.25 were considered as having *"nothing in common"*, between 0.25 and 0.5 as being *"somewhat related"*, between 0.5 and 0.75 as being *"rather similar"* and between 0.75 and 1.0 as being *"very close"*.



(a) (b)

Fig. 3. Results (a) before and (b) after interval adjustment

The system appears more optimistic than users. This is apparently due to the max functions in our measure and to the fact that it does not know the context and may consider some words as semantically close even if the user disagrees. If we except the sentences having *"nothing in common"*, we can notice that configurations ♯1 and ♯4 have a behavior somewhat similar to that of the user.

We focused on these two candidates. To assess their quality, we refined our results – without changing user judgment – by redefining the intervals representing the similarity appreciation. Figure 3 (b) shows the results obtained for [0.0, 0.39],]0.39, 0.58],]0.58, 0.72] and]0.72, 1.0]. ♯1 and ♯4 did then match the appreciation of the users rather well. To check that they really were good parameters, we separately considered the results for the different ontologies. These are shown by figure 4.

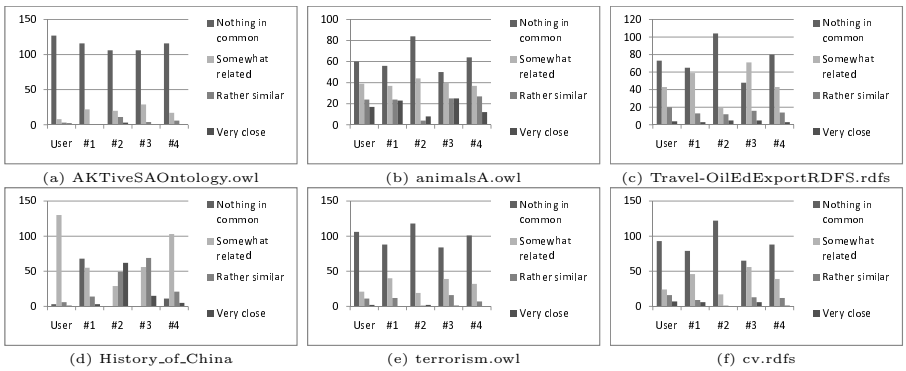


Fig. 4. Results after interval adjustment for each test ontology

Figure 4 (a) and (b) show that both configuration ♯1 and ♯4 behave well with different numbers of topics. The fact to not consider the overall semantics does not seem to play a too important role.

Results presented by (c) and (d) are more interesting, since they show a strong difference mainly due to vocabulary complexity. While ♯1 and ♯4 behave rather well with low term complexity, ♯1 appears to be more unstable than ♯4 when the system is confronted to unknown words. This indicates the good potential of ♯4.

Finally, (e) and (f) show that our measure is generally not affected by the internal complexity of the ontologies, and comfort the choice of ♯4, which mimics rather well human judgment. This comfort also our idea to consider all aspects of the triples and not only subjects, since ♯1 has more weight on subject-subject match than ♯4.

5.2 Snippet Quality

Using configuration ♯4, we considered three test activities. Users had to first find the best threshold for the topics. They were then asked to evaluate the quality of the clustering. Finally, we asked them to rank topics and sentences related to a query and compared their result with that of the system.

To find the best threshold, we choose three small ontologies and divided our users into groups of three. Each group was given an ontology. Each person worked alone and was provided a screen capture of the RDF graph obtained with a visualization plugin of Protégé[9]. After studying the ontology, they were asked to use the upload interface of our demo system repeatedly and to look for the threshold providing the best clustering. Their answers varied between 0.65 and 0.75, with an average of 0.71.

The quality of the clustering itself was assessed by a questionnaire. Testers should indicate on a four-level scale if the topics were conformed to the choice they would have made, if they estimated the topics to be coherent and what they thought about the overall quality of the clustering. A majority of users (5 out of 9) considered the result as conform to what they would have done. Others considered that they would probably have made some adjustments. The coherence of the topics gave approximately the same result, while everybody agreed on a good quality of the clustering, two users even considering it as very good.

It appeared that even with some slight clustering differences, the result of the biased snippets matched rather well the selection made by the users.

5.3 Performance

Performance for different ontologies is given by figure 5. We can notice the important time needed by the clustering phase w.r.t. the size. This justifies the off-line step but will also encourage further optimizations. The comparison between `cv.rdfs` and `History_of_China` points out the role of the vocabulary complexity. The more distinct words an ontology contains, the more access to the WordNet database are needed, which slows down both online and off-line

[9] `http://protege.stanford.edu/`

Ontology	Size	Triples	Sentences	Clustering	Total	Snippet
Cat_health.rdf	5 KB	25	22	3.484 s	4.171 s	0.141 s
animalsA.owl	8 KB	129	89	7.156 s	7.547 s	0.515 s
cv.rdfs	23 KB	419	248	17.03 s	18.343 s	1.110 s
History_of_China	52 KB	275	272	74.433 s	75.293 s	2.531 s
terrorism.owl	188 KB	2382	1438	737.472 s	754.156 s	0.187 s

Fig. 5. Clustering time, total pre-processing time and snippet generation time.

steps. It appeared during our experiments that the time of the online module mostly depends on disk and network access speed. The fact to save the pre-processed snippet and to load it again represents up to a few seconds.

5.4 User Feedback

Our testers were asked some questions and were free to leave comments. While they all agreed on the readability and accuracy of the snippets, two users were not fully convinced, even if they did not deny the advantages brought by the system. One of them proposed to further investigate with a larger set of users and documents, which is in our opinion an interesting further step for our work along with its integration into a real search engine. Others were rather pleased with the potential of the proposed solution.

6 Related Work

To the best of our knowledge, query-biased snippet generation from ontologies is still a largely unexplored field. [12] recently proposed a solution to generate snippets based on term occurrence. Contrary to our approach, topic identification or similarity measure were not considered. Without precisely considering the case of snippets, [13] and [14] investigated ontology evaluation and selection, and illustrated the importance and the openness of this issue. Snippet generation is also not limited to ontologies and is still actively discussed. [9] proposed strategies to increase performance of snippet generation through document caching, which are interesting for further optimize our system, since disk storage and access have shown to cost up to a few seconds per snippet.

Document summarization and clustering are two fields closely related. Work on multiple text documents summarization, like [15] that use a centroid-based approach for topic determination, comes now along with ontology summarization like [2]. Our extension of the later with the addition of topics and topic graphs is inspired by techniques used in text summarization. Ranking ontological structure is also required to generate snippets of different length that still contain the essence of the ontology, as discussed by [2]. While we reuse their results, we also include the possibility to bias the ranking results according to the query provided by the user.

Similarity between concepts plays an important role in domains such as Information Retrieval, Natural Language Processing or even Genetics [16], and has been studied a lot in the literature. It also aims at facilitating ontology merging and aligning. [4] proposes an approach to compare ontology structures both from the conceptual and the lexical point of view. Even if we shared this idea, we did not separate both aspects like they did, since we considered structure and meaning to be closely related within the particular structure of RDF sentences. [8] has considered the similarity from a theoretical point of view and was an interesting methological help in the design process of our measure.

Different methods to measure semantic similarity between words are investigated by [7]. To outperform the traditional edge counting approach, [6] proposed a semantic similarity measure between terms using WordNet. Inspired by this work, [3] describes a new measure, used by our system[10]. However, its limitation to nouns and verbal bases brought us to add the possibility to consider adjectives, adverbs and conjugation as well. [17] designed a measure to compare different ontological structures, which differs from our approach since we do not only consider class names.

Finally, our system includes some characteristics of NL paraphrasing of ontologies, as investigated in [10] and proposed by [11]. The ideas described by this related work on NL go far beyond what we decided to implement but show what can be achieved in a near future.

7 Conclusion and Future Work

In this paper, we proposed a solution to the problem of ontology selection for non-specialists. Our system relies on a new semantic similarity measure, that exploits the semantics of structures called RDF sentences, does not limit its scope to class or entity names, and considers all words in the ontology resources without being limited to nouns and verbs. It also extends and gives an application to works like [2] and [3], and introduces a new and friendly way to consider results returned by semantic web search engines. A user evaluation assessed its potential and highlighted a few tracks for further development.

The next steps involve improvements in the clustering process, since our test data shows that it takes most of the running time. It is also planned to improve the natural language results, to be closer to the ideas expressed in [11]. The main future achievement will be the inclusion of the system into a real search engine, to further improve user support and to benefit from information such as relevance score computed by the engine.

References

1. Battle, L.: Preliminary inventory of users and tasks for the semantic web. 3rd Intl. Semantic Web User Interaction Workshop (2006)

[10] `http://eden.dei.uc.pt/~nseco/javasimlib.tar.gz`

2. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on rdf sentence graph. Proceedings of the 16th international conference on World Wide Web (2007)
3. Seco, N., Veale, T., Hayes, J.: An intrinsic information content metric for semantic similarity in wordnet. Proceedings of 15th ECAI (2004)
4. Maedche, A., Staab, S.: Measuring similarity between ontologies. Proceedings of the European Conference on Knowledge Engineering and Knowledge Management (2002)
5. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. International Semantic Web Conference (2005)
6. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artifi cial Intelligence Research 11 (1999)
7. Petrakis, E.G., Varelas, G., Hliaoutakis, A., Raftopoulou, P.: X-similarity: Computing semantic similarity between concepts from different ontologies. Journal of Digital Information Management (JDIM) (2006)
8. Lin, D.: An information-theoretic definition of similarity. Proc. 15th International Conf. on Machine Learning (1998)
9. Turpin, A., Tsegay, Y., Hawking, D., Williams, H.E.: Fast generation of result snippets in web search. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (2007)
10. Hewlett, D., Kalyanpur, A., Kolovski, V., Halaschek-Wiener, C.: Effective nl paraphrasing of ontologies on the semantic web. Workshop on End-User Semantic Web Interaction, 4th Int. Semantic Web conference, Galway, Ireland (2005)
11. Wilcock, G.: Talking owls: Towards an ontology verbalizer. Proceedings of the ISWC Workshop on Human Language Technology for the Semantic Web and Web Services (2003)
12. Huang, Y., Liu, Z., Chen, Y.: Query biased snippet generation in xml search. Proceedings of the ACM SIGMOD International Conference (2008)
13. Sabou, M., Lopez, V., Motta, E., Uren, V.: Ontology selection: Ontology evaluation on the real semantic web. Proceedings of WWW (2006)
14. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept structures. Proceedings of the 3rd international conference on Knowledge capture (2005)
15. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing and Management 40 (2004)
16. Couto, F.M., Silva, M.J., Coutinho, P.M.: Semantic similarity over the gene ontology: Family correlation and selecting disjunctive ancestors. Proceedings of the CIKM Conference (2005)
17. Rodriguez, M.A., Egenhofer, M.J.: Determining semantic similarities among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering (2003)