Discovering the Semantics of Keywords: An Ontology-based Approach*

Mauricio Espinoza†, Jorge Gracia, Raquel Trillo, and Eduardo Mena

IIS Department, Univ. of Zaragoza
María de Luna 1, 50018 Zaragoza, Spain
E-mail: {mespinoz,jogracia,raqueltl,emena}@unizar.es

Abstract

In the context of the emerging Semantic Web, a great effort has been done in the construction of ontologies. An increasing number of them is becoming available on the Web, in order to share the knowledge that they represent.

In this paper we propose an automatic mechanism that accesses, extracts and semantically merges the knowledge contained in a pool of ontologies available on the Web. In particular, we have focused on the problem of discovering the set of candidate meanings for a given keyword (or keywords). First, the different senses are obtained from different ontology libraries; second, redundant senses are automatically integrated when the system determines that they can be considered synonyms; the previous two steps are repeated until all the ontologies are visited or a specified amount of time is spent.

This method proposed to manage keyword senses can be used for different purposes, such as annotation of web pages, keyword sense disambiguation, ontology matching, etc.

Keywords: Semantic interoperability, semantic web mining, data semantics

1 Introduction

According to [8], ontologies are specifications of conceptualizations. They play a fundamental role in the Semantic Web [3] because they are used to make the semantics of terms explicit without ambiguity. Thus users can create and express different concepts[1] by defining ontological terms with the purpose of adding semantics to the information available on the Web.

Nevertheless, creating an ontology from scratch usually requires a great effort (and the intervention and agreements of human experts). Besides building new ontologies, the growth of the Semantic Web needs the sharing of existing ontologies in an efficient way.

Currently the access to pre-existing ontologies is not a difficult task due to the increasing of ontology libraries, such as SchemaWeb [17], and the availability of ontology searchers as Swoogle [6]. However the use of such tools is still limited to obtaining a list of potentially related ontologies with no further support to extract the required knowledge. Just finding related ontological terms in a multiontology environment is not useful enough for some tasks (e.g. keyword sense disambiguation), so it would be desirable a mechanism that semantically merges the knowledge coming from different ontologies when its semantics is similar enough.

In this paper we propose a system that takes as input a keyword (the same process should be repeated for a set of keywords) and retrieves its possible semantic meanings according to the knowledge extracted from a third party ontology pool. The main steps are: 1) The system matches the keyword against the terms of ontologies in the pool using Swoogle and other lexical resources; 2) the system merges those senses considered very similar, by computing the synonymy probability between them; otherwise it considers them as different senses; the result is a set of independent possible senses for the keyword entered. Our approach uses sampling and other statistical techniques, as well as parallel processing, whenever possible, in order to improve the efficiency of the system. We also take advantage of the shared knowledge and semantic web services like Swoogle available in the web.

Our system has been developed as a part of a broader project (described in [7]), which is mainly focused on keyword disambiguation. However, the same mechanism can be used with other purposes, like obtaining a set of relevant ontological senses (for example, for a semantic annotation).

The structure of the rest of this paper is as follows. In Section 2 we describe the architecture of our system. In Section 3 we describe how to extract the candidate senses

*Work supported by the CICYT project TIN2004-07999-C02-02.

†This work is supported by a grant of Santander Central Hispano & University of Zaragoza.

[1] We use the terms concept, role and value with the same meaning as class, property and instance in OWL [2], respectively.

from an ontology pool. In Section 4 we describe the algorithm that computes the synonymy degree between two candidate senses, and integrates them when a certain threshold is achieved. Section 5 presents some related works. Finally, conclusions and future work appear in Section 6.

2 Overview of the system

In our opinion, to achieve our goal, several requirements should be considered: 1) the system must deal with distributed and heterogeneous ontologies elaborated by experts in different domains, 2) the system should access to ontologies available on the Web in order to avoid local maintenance and to improve flexibility, 3) ontologies must be described in knowledge description languages as OWL, DAML or RDF [2], and 4) the system should avoid redundancy across ontologies, integrating knowledge that represent the same concept. In the following we summarize the main steps given by our approach to obtain a set of independent keyword senses for a user keyword (see Figure 1).

1. Keyword Normalization: In order to facilitate the syntactical comparison between keywords and ontology terms, a syntactical normalization is performed. It is carried out by rewriting the keywords in lowercase, removing hyphens and other special characters, and performing a morphological processing (plural names are transformed in single names; verbal forms are put in infinitive, etc).

2. Extraction of Senses: The normalized keywords are searched among the terms of an ontology pool. The use of this ontology repository increases the probability of discovering the most suitable sense for each keyword, as many interpretations are described in those third-party ontologies. We access to ontologies by means of Swoogle [6], a system that indexes many ontologies available on the Web. Each ontology term indexed is used to browse the corresponding ontology; a black list is managed to deal with ontologies whose quality is very low, and a buffer of previously parsed ontologies us used to avoid parsing the same ontology twice. Each new ontology terms is explored and a new candidate sense is build taking its ontological context (hypernyms, hyponyms) as basis. A detailed description of this process is presented in Section 3.

3. Computation of synonymy degree and integration of senses: The new candidate sense of the keyword should be compared with the previously found senses (which are obtained incrementally, as we can see in Figure 1). For this task, the system computes the synonymy probability between the candidate sense and each previously obtained sense. If the synonymy degree is above a certain threshold, the system considers both senses synonyms, integrates them by preserving the semantic context of both, and updated the list of senses fond. In other case, the candidate sense represents a new semantics for the keyword, and is stored as a new possible sense. This step is explained in detail in Section 4.

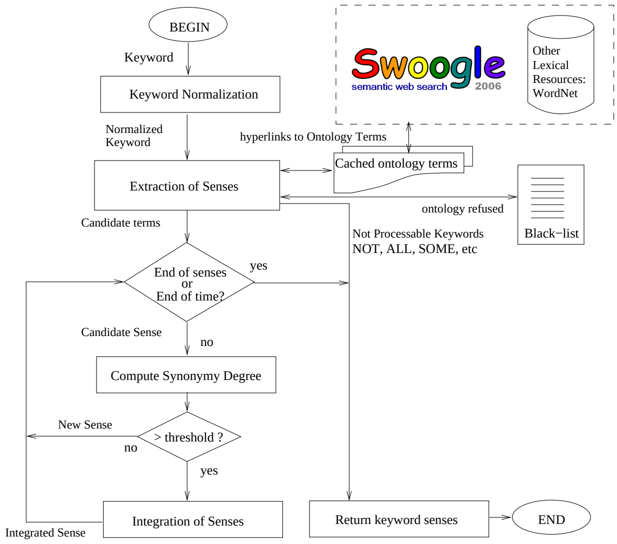

Figure 1: Main steps

3 Extraction of Senses

Our system searches semantic resources in existing knowledge sources by means of Swoogle, a system that allows keyword-based search for online ontologies, and indexes about 10,000 ontologies available on the Web. It indexes semantic documents that contain Semantic Web Terms (SWT), which play the role of words in natural languages. SWTs bridge RDF statements with formal semantics defined in RDF(S) and OWL, and are intended to be reused as universal symbols [5]. Also local ontologies (not indexed by Swoogle) and other lexical resources (as WordNet [12] can be loaded and utilized as well.

As motivating example, let us suppose that a user is interested in defining the concept *developer* (for example for semantic annotation of web pages, text disambiguation, etc.), and his intended sense is software developer. When we enter the keyword *developer* in a lexical resource as WordNet, only two senses are retrieved: 1) someone who develops real estate (especially someone who prepares a site for residential or commercial use), and 2) photographic equipment consisting of a chemical solution for developing film. However, Swoogle returns six ontology terms: 1) a property that store the developer of software for the project (which matches the user intended meaning), 2) subclass of creator, 3) developer as subclass of member of a project (the developer of the project), 4) someone or a group of people who develop something, 5) someone or a group of people who develop something (same semantics as before), and 6) photographic equipment consisting of a chemical solution for developing film (which is the same meaning as the first sense retrieved from WordNet). We can see that the more ontologies or knowledge bases accessed, the more chances to find the semantics that the user is looking for. In our approach we look for keyword senses not only accessing to Swoogle but also other high quality lexical resources as WordNet.

As result of a Swoogle search a list of links (URIs) to semantic web terms (that syntactically match the searched keyword) is returned. Each ontology is rated by taking into account its *quality level*[2], and so high-rated ontologies are consulted first. As Swoogle could return ontologies that contain syntax errors, or outdated URIs[3], only those ontologies free of error when parsed are taken into account; otherwise they are annotated in a black list of ontologies. A cache mechanism has been implemented to avoid parsing previously analyzed ontologies twice.

Obtaining Candidate Senses

For those error-free ontologies with ontology terms that match the user keyword, the system starts the process of extraction of senses taking the matching ontology terms as basis. In this paper, a sense of a keyword k, denoted by s_k, is a tuple s_k = <*s, grph, descr, pop, syndgr*>, where s is the list of synonym names of keyword k, $grph$ describes the sense s_k by means of the hierarchical graph of hypernyms and hyponyms of synonym terms found in one or more ontologies, $descr$ is a description in natural language of such a sense (if available), and pop and $syndgr$ measure the degree of popularity of this sense (number of times it appears in the ontology pool) and the integrated percentage of synonymy degree, respectively; pop and $syndgr$ are used when this sense integrates knowledge from different sources. For a better understanding we provide some sample senses later.

As matching terms could be ontology concepts, roles, or values, three lists of possible senses are associated with each keyword k: $S_k^{concept}$, S_k^{role} and S_k^{value}, to store the senses that k can play as concept, role, or value, respectively. The first matching becomes the first found sense for that keyword.

In order to recover the sense description in which we are interested, we used the technique of extraction by traversal [13] applied to concepts, roles or instances. The central ontological term of interest, and the relationships to traverse to find other terms, are calculated automatically for the system. The depth value for the traversal of $grph$ (to which we call ontological context), depends on the ontology specification level and required quality for the description of a sense, but by experimentation we found that low values provide good results.

We distinguish two depth levels: depth context definition and depth context matching. The first level is useful to show the sense to the user. On the other hand, the depth context matching is used to calculate the synonym degree between two senses extracted from different ontologies. By default these two levels have the same value, but is possible to change them to improve the quality of synonymy degree computation without complication the visual representation of the sense.

As matching terms could be ontology concepts, roles, or values, the ontological context in each case is different. As we said before, in the example the system finds seven matchings of keyword *developer* as concept/class, one as role/property, and no occurrence of "developer" as value/instance. Thus, those eight senses constitute the *candidate senses* found for the keyword "developer". We show in Figure 2 only the first three senses found, due to space limitations. Each sense is initialized with a popularity=1 (first occurrence) and a degree of synonymy=1.

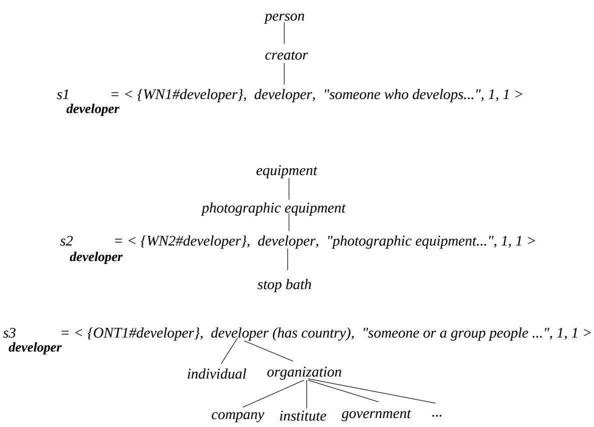

Figure 2: Three first senses of "developer"

In the following, we explain how the first sense shown in Figure 2 was extracted from its ontology (http://xmlns.com/wordnet/1.6/Developer). The concept *developer* is the "focus" from which the system retrieve the ontological context to the specified depth.

In Figure 3, the solid circle represents the depth context definition (value=2) and the depth context matching is represented by the dotted circle (value=3). As we explained previously, these values could be the same. The concepts in bold constitute the definition of the sense, the other ones are used to compute the *probability synonym* between this candidate sense and the rest of candidate

[2]Manually assigned to those ontologies whose (good or bad) quality is known, from the point of view of semantic agreement among knowledge engineers.

[3]This is the price to pay for indexing ontologies available on the Web.

senses. The set of synonyms[4] is specified between brackets and the description in natural language (when available) appears between square brackets . In this particular example, the term “developer” has no hyponym or specific role (range and cardinality are extracted in the case of role senses).

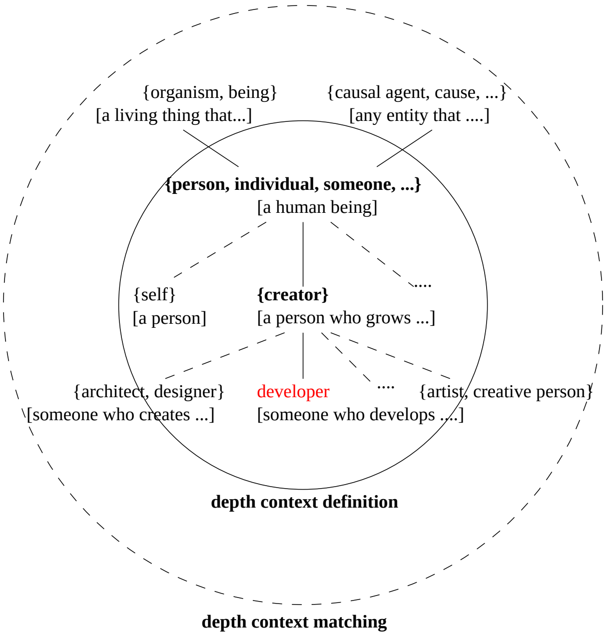

Figure 3: Extraction of ontological context of sense of “developer”

In summary, if a keyword match an ontological term thatis as concept, the system extracts the set of synonyms, hypernyms and hyponyms (to the specified depth following the ontology graph), roles, and its description in natural language (when available). When the keyword matches with an ontology role, a similar procedure is performed on the role ontological context[5]. The information extracted in this case is the set of superroles, subroles, domains, ranges and any other roles restriction available. If the keyword matches with an ontology value, the system extracts the ontological context of the concept that value belongs to.

4 Synonymy Probability between keyword senses

Different ontologies often semantically overlap by defining similar or somehow related terms. In our system we compute the probability of synonymy between two candidate senses to conclude whether they are semantically the same sense or not. Thus the system avoids redundancy among the possible keyword senses. The Figure 4 details how the algorithm that computes the synonymy degree works.

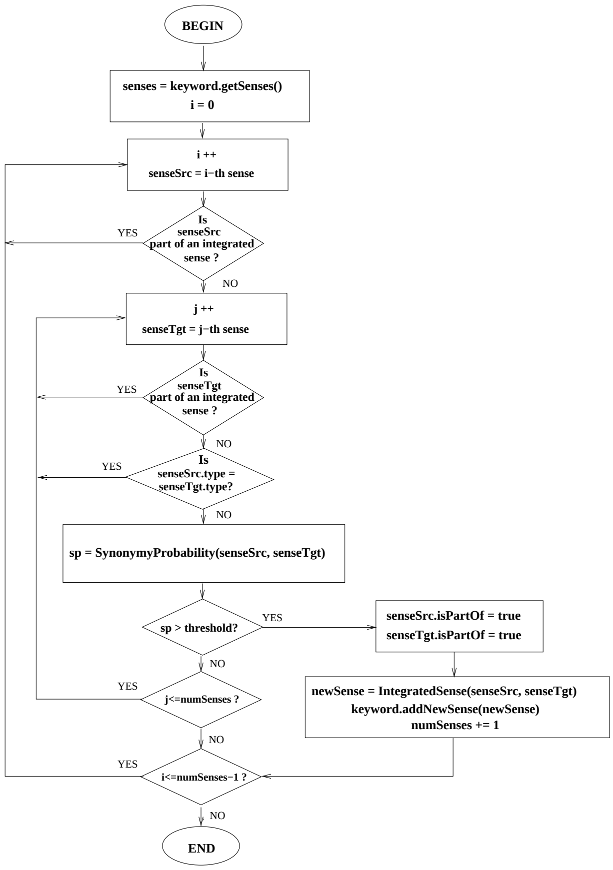

Figure 4: Synonymy degree computation

Notice that the system does not compare those senses coming from the same ontology; we suppose that different occurrences of a term in the same knowledge source always describe different senses if that ontology is free of redundancy, as expected in a well-formed ontology. We would like to point out that the system estimates the synonymy probability using a statistical approach, as show in [15], and incrementally calculates when two terms are equivalent by studying a sample of the hypernym/hyponym relationships. This sampling method improves the performance of the systems as just a few comparisons are performed to compare two senses. We so adapt the idea of [15] (which only works for WordNet) to a multiontology context.

The way to obtain the synonymity probability varies according to the type of senses that we compare. The expressions $sp_{concept}$, sp_{role} , and sp_{value} compute the synonym probability in each case. In the following we

[4] In order to extract from an ontology the synonyms of a concept, role or value, we looked for the primitives *equivalentClass*, *equivalentProperty* and *sameIndividualAs* that define equivalences between concepts, roles, and values, respectively. WordNet provides this information together with the term definition

[5] Ontology roles can be organized in hierarchies of roles

explain how the synonymy probability between a previous sense s_{t1} and a new one s_{t2} is computed, whether $t1$ and $t2$ are concepts, roles, or values.

4.1 Synonymy probability between concepts

We propose a synonymy probability function $sp_{concept}$ that is defined by the weighted sum of 1) $l(t1, t2)$, the lexicographical distance [10] between concept names $t1$ and $t2$, 2) sp_{ctx} , the similarity between the hypernym and hyponym hierarchies (let us call them *contexts*) of $t1$ and $t2$, with a certain depth d, 3) sp_{rSet}, the similarity between the role sets of concepts $t1$ and $t2$ and 4) sp_{des}, the similarity between the sense descriptions.

$$
\begin{aligned}
sp_{concept}(s_{t1}, s_{t2}, d) = {} & w_{name} \cdot l(t1, t2) \\
& + w_{ctx} \cdot sp_{ctx}(s_{t1grph}, s_{t2grph}, d) \\
& + w_{roles} \cdot sp_{rSet}(s_{t1}, s_{t2}, d) \\
& + w_{des} \cdot sp_{des}(s_{t1descr}, s_{t2descr})
\end{aligned}
$$

$$
\begin{gathered}
w_{name}, w_{ctx}, w_{roles}, w_{des} \geq 0 \\
\wedge w_{name} + w_{ctx} + w_{roles} + w_{des} = 1
\end{gathered}
$$

The synonymy probability sp_{ctx} is based on the recursive formula presented in [15]. It follows a statistical approach to incrementally calculate the degree of similarity between two terms by analyzing a *sample* of the hypernym/hyponym relationships between terms, with a certain depth; sampling techniques are used to improve performance in case of ontologies are very populated. This function never returns high values for "bad" synonyms. More details can be found in [15]. However, we adapted such a formula to calculate the synonymy probability between two senses belonging to terms in different ontologies. Thus, if the synonymy probability is beyond a threshold, the two senses will be considered the same. Other semantic similarity measures can be used, see Section 5.

$$
sp_{ctx}(s_{t1}, s_{t2}, d) =
\begin{cases}
\frac{\sum_{s_{ti} \in h_{s_{t1}}}^{s_{tj} \in h_{s_{t2}}} l(s_{ti}, s_{tj})}{|h_{s_{t1}}||h_{s_{t2}}|} + \frac{\sum_{s_{ti} \in H_{s_{t1}}}^{s_{tj} \in H_{s_{t2}}} l(s_{ti}, s_{tj})}{|H_{s_{t1}}||H_{s_{t2}}|} & if\ d = 1 \\
cd\left(\frac{\sum_{s_{ti} \in h_{s_{t1}}}^{s_{tj} \in h_{s_{t2}}} l(s_{ti}, s_{tj})}{|h_{s_{t1}}||h_{s_{t2}}|} + \frac{\sum_{s_{ti} \in H_{s_{t1}}}^{s_{tj} \in H_{s_{t2}}} l(s_{ti}, s_{tj})}{|H_{s_{t1}}||H_{s_{t2}}|}\right) + & \\
+(1 - cd) \cdot hH(s_{t1}, s_{t2}, d) & otherwise
\end{cases}
$$

$$
hH(s_{t1}, s_{t2}, d) = \\
= \frac{\sum_{m \in h_{s_{t1}}}^{n \in h_{s_{t2}}} sp_{ctx}(m, n, d-1)}{|h_{s_{t1}}||h_{s_{t2}}|} + \frac{\sum_{m \in H_{s_{t1}}}^{n \in H_{s_{t2}}} sp_{ctx}(m, n, d-1)}{|H_{s_{t1}}||H_{s_{t2}}|}
$$

$h_{s_{t1}}$, $h_{s_{t2}}$ are the *sampled set* of hyponyms of the hierarchical graph in senses s_{t1} and s_{t2}, respectively; $H_{s_{t1}}$, $H_{s_{t2}}$ are the sampled hypernyms, $|X|$ is the cardinality of the set X, and cd represent the *certainty degree* (see [15]) for obtaining the sample size for the four sets of terms: $h_{s_{t1}}, h_{s_{t2}}, H_{s_{t1}}$, and $H_{s_{t2}}$. The function $hH(m, n, d)$ calculate synonymy probability of the context (hypernyms and hyponyms) of the ontological terms m and n, with depth d.

The synonymy probability sp_{rSet} between the role set of two concept senses s_{t1}, and s_{t2} is computed in the next way:

$$
sp_{rSet}(s_{t1}, s_{t2}, d) = \frac{\sum_{m \in roles_{s_{t1}}}^{n \in roles_{s_{t2}}} sp_{role}(m, n, d)}{|roles_{s_{t1}}||roles_{s_{t2}}|}
$$

where, $roles_s$ denotes the role set of ontology terms belonging to sense s and $sp_{role}(r1, r2, d)$ estimates the synonymy probability of two ontology roles, with a certain depth d (explained in the following).

4.2 Synonymy probability between roles

To calculate the synonymy probability between roles $sp_{role}(r1, r2, d)$, we integrate the information obtained from the semantic context of roles[6] $r1$ and $r2$ and the semantic context of their domain and range concepts.

$$
\begin{aligned}
sp_{role}(r1, r2, d) = {} & w_{ctx} \cdot sp_{ctx}(s_{r1}, s_{r2}, d) \\
& + w_{domain} \cdot sp_{concept}(s_{domain(r1)}, s_{domain(r2)}, d) \\
& + w_{range} \cdot sp_{concept}(s_{range(r1)}, s_{range(s2)}, d)
\end{aligned}
$$

$$
\begin{gathered}
w_{ctx}, w_{domain}, w_{range} \geq 0 \\
\wedge w_{ctxR} + w_{domain} + w_{range} = 1
\end{gathered}
$$

where w_{ctx}, w_{domain}, and w_{range} are the weights of the above synonymy measures, and $domain(r)$ and $range(r)$ return the domain concept and range concept of role r, respectively.

4.3 Synonymy probability between values

To compute the synonymy probability $sp_{value}(v1, v2, d)$ between two constant values of different ontologies, the synonymy of the concept they belong to, with a certain depth d.

$$
sp_{value}(v1, v2, d) = sp_{concept}(s_{concept(v1)}, s_{concept(v2)}, d)
$$

where $concept(v)$ returns the concept from which v is an instance.

Concerning our example, the probability of synonymy between each pair[7] of candidate senses are shown in Table 1 (depth=2 in our prototype).

Thus the system obtains a list of ontological terms (senses) that can be integrated: those senses whose synonymy probability is greater than the specified threshold (0.7 in our prototype) (indicated in bold in Table 1).

[6] Ontology roles can be organized in hierarchies of roles.

[7] Symbols and empty cells represent that the comparison makes no sense due to different reasons.

Table 1: Synonymy probability among senses

sens.	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
s_1	◁							
s_2	-	◁						
s_3	-	-	◁					
s_4	0.56	0.52	-	◁				
s_5	0.56	0.52	-	**0.93**	-			
s_6	0.54	**0.84**	-	*		-		
s_7	0.63	*	-				◁	
s_8	0.53		-				0.24	◁
s_2+ s_6			-				0.45	0.26
s_4+ s_5							0.46	0.28

4.4 Integration of Senses

When the synonymy probability between two senses is greater than a threshold, the system considers them as the same sense and integrates $s1_{keyword}$ and $s2_{keyword}$ into one only sense. Otherwise, the $s2_{keyword}$ is stored in $S^{concept}_{keyword}$ as a new sense. In this way, new knowledge found about a keyword is added to, or integrated with, the previously obtained. Also, knowledge discovered is used in future comparisons to avoid recalculations.

The integration process starts by adding to the synonym set (s) the union of the synonyms of both senses; similarly the two sets of hyponyms and hypernyms are set as hyponyms and hypernyms of the integrated term, respectively. The two descriptions in natural language ($descr$) are just concatenated. The value of popularity (pop) is updated as the sum of its source values and the synonymy degree (syn) is the value obtained by the computation of the algorithm explained in the previous section.

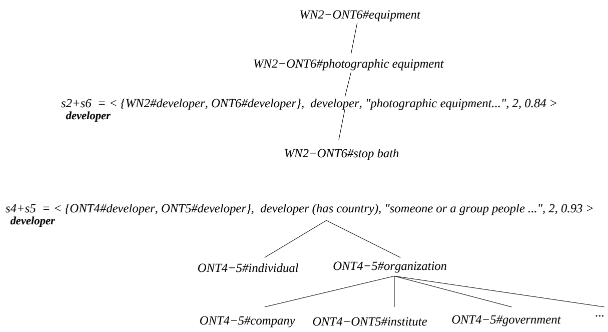

Figure 5: Integration of senses s2 and s6, and integration of s4 and s5

Concerning the example, in Figure 5 we show two pairs of senses, whose synonymy probability is greater than the threshold and therefore they are integrated. In the example the senses $s2$ and $s6$ are integrated in $s2 + s6$, and $s4 + s5$, represents the integration of senses $s4$ and $s5$. Notice that new popularity is 2 in both cases, and the degree of synonymity of $s2+s6$ is 0,84 and the one of $s4+5$ is 0,93. Ontological contexts has been also integrated.

5 Related Work

In [1] they suggest exploiting the functionality provided by the available Semantic Web tools and techniques. They identify a set of tools for each phase of ontology construction, but do not specify the way to utilize them together. For example, they propose some techniques of segmentation of ontologies, but they do not identify what information to extract (concepts, roles or instances) and how it will be used later. Also, unlike this work, we look for ontological terms that matching with keyword search.

In [16] they use the domain knowledge inherent in Web-based classification hierarchies such as Yahoo, to enhance user queries. This work do not take advantage of other semantic relationships found in a domain-specific ontology. Our work can be utilized for this purpose, with inherent advantages to query a multiontology knowledge base. In [9] they search concepts over all the ontologies in a local repository. We avoid local maintenance of ontologies and increase flexibility by accessing to the last updated versions of on-line ontologies through Swoogle.

In order to segment large description ontologies some techniques exist currently. We used the concept of *Traversal View*, presented in [13] for this purpose. A user specifies a subset of an ontology to include in the view by specifying the starter concepts to include, the links to follow from those concepts, and the search depth. However, we extend this use to the extraction of ontological terms that are roles (properties) or values. Other techniques as the use of queries to create specific ontology views [11] could be used as well.

An alternative measure to estimate the semantic similarity between terms in different ontologies can be found in [14]. Instead of our recursive approach, they compare entity classes considering three independent similarity assessments: synonym sets, distinguishing features and semantic neighborhoods, combining them. This method is very suitable for comparisons among ontologies where many different semantic relationships have been defined across terms (which, unfortunately, happen in just a few ontologies, as WordNet). Furthermore they do not describe how a comparison between roles or between values can be performed.

Finally, some efforts oriented to publishing and searching ontologies on the Web, like DAML Ontology Library [4], SchemaWeb [17], Swoogle [6], contribute to our proposal as resources for searching ontologies; however we go further on extracting ontology parts in which we are interest and discovering semantic similarity among terms of different ontologies.

6 Conclusions and Future Work

In this paper we have presented an approach to discover senses for a set of user keywords, extracting the relevant ontological contexts from ontologies with matching terms and merging senses when the synonym probability between them is high enough. The main features of our proposal are the following:

1. It uses a multiontology repository to search candidate keyword senses, instead of consulting just one lexical resource.

2. It has been designed to exploit the functionality provided by the Semantic Web existing available tools and techniques.

3. It uses statistical techniques like sampling to improve the performance of the synonymy degree computation. Comparisons between senses can be also executed in parallel.

4. We deal with senses represented by concepts, roles and values, defining specific synonym degree computations and integration techniques for each case.

Also our prototype can be used through a Web navigator, without requiring the installation of any other software.

As future work we could adopt the proposal presented in [9] to provide a natural language explanation of senses. Also, we are currently using this work to disambiguate sets of keywords and we plan to use it for web mining.

References

[1] H. Alani. Position paper: Ontology construction from online ontologies. In *15th International World Wide Web Conference (WWW2006), Edinburgh, UK*, May 2006.

[2] G. Antoniou and F. van Harmelen. Web ontology language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*. Springer-Verlag, 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientic American*, May 2001.

[4] DAML Ontology Library., 2003. http://www.daml.org/ontologies/.

[5] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and ranking knowledge on the semantic web. In *Proceedings of the 4th International Semantic Web Conference*, November 2005.

[6] T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the Semantic Web. In *AAAI 05 (intelligent systems demo)*, July 2005.

[7] J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the web: A multiontology disambiguation method. In *Sixth International Conference on Web Engineering (ICWE'06), Palo Alto (California, USA)*. ACM, July 2006.

[8] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, 1993.

[9] A. Kalyanpur, N. Hashmi, J. Golbeck, and B. Parsia. Lifecycle of a casual web ontology development process. In *Proceedings of the Workshop on Application Design, Development and Implementation Issues in the Semantic Web, New York (New York, USA)*, May 2004.

[10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1966. Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848 (1965).

[11] A. Magkanaraki, V. Tannen, V. Christophides, and D. Plexousakis. Viewing the semantic web through rvl lenses. In *The Second International Semantic Web Conference (ISWC'03)*, 2003.

[12] G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), nov 1995.

[13] N. F. Noy and M. A. Musen. Specifying ontology views by traversal. In *Proceedings of International Semantic Web Conference, ISWC04*, November 2004.

[14] M. Rodriguez and M. Egenhofer. Determining semantic similarity among entity classes from different ontologies, 2003.

[15] J. Royo, E. Mena, J. Bernad, and A.Illarramendi. Searching the web: From keywords to semantic queries. In *Third International Conference on Information Technology and Applications (ICITA'05), July 4-7, 2005, Sydney (Australia)*, pages 244–249. IEEE Computer Society, ISBN 0-7695-2316-1, July 2005.

[16] A. Sieg, B. Mobasher, S. Lytinen, and R. Burke. International conference on artificial intelligence and applications. In *Using Concept Hierarchies to Enhance User Queries in Web-based Information Retrieval, Innsbruck (Austria)*, February 2004.

[17] VicSoft., 2003. http://www.schemaweb.info/.