A Semantic Method for Searching Knowledge in a Software Development Context*

Sonia Bergamaschi, Riccardo Martoglia, and Serena Sorrentino

DII, University of Modena and Reggio Emilia
via Vignolese 905, 41125 Modena, Italy
firstname.lastname@unimore.it

Abstract. The FACIT-SME European FP-7 project targets to facilitate the use and sharing of Software Engineering (SE) methods and best practices among software developing SMEs. In this context, we present an automatic semantic document searching method based on Word Sense Disambiguation which exploits both syntactic and semantic information provided by external dictionaries and is easily applicable for any SME.

1 Introduction and Motivation

Over the last years, in Europe, software development is becoming a bottleneck in the development of the Information Society, especially for SMEs (Small and Medium Enterprises) which need to allocate mostly all of their available resources on its production rather than on new technology training. The main goal of the European FP7 3 years project "Facilitate IT-providing SMEs by Operation-related Models and Methods (FACIT-SME)" is to facilitate IT SMEs in sharing and (re)using SE methods, tools, and experiences for systematically designing and developing their applications integrated with the business processes. In order to achieve this goal, the project proposes a novel Open Reference Model (ORM) [4] serving as an underlying knowledge backbone which stores existing reference knowledge for software-developing SMEs, including different engineering methods, tools, quality model requirements, and enterprise model fragments of IT SMEs in a computer-processable form. On top of the ORM repository, a customizable Open Source Enactment System (OSES) [3] provides IT support for the project-specific application of the ORM. As key part of the OSES, specific query-based search methods support the organizations in: finding a new methodology, by selecting ORM elements that best match given specific enterprise objectives (i.e., "From Scratch" scenario); improving a given existing methodology, suggesting the ORM information most relevant to it (i.e., "From methodology" scenario).

* This extended abstract summarizes the research work we performed in the first two years of the FACIT-SME project, including a summarization of the preliminary results described in [12] (SEKE 2011). It was partially supported by the European Community's Seventh Framework Programme managed by REA Research Executive Agency (http://ec.europa.eu/research/rea)([FP7/2007-2013][FP7/2007 - 2011])

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

In our research work, we focus on search/filtering methods by taking advantage of the textual information (which we will refer to as *documents*) stored in the ORM and/or already available in each enterprise. In this context, queries are provided in textual form, e.g. keywords/sentences about the company background and project requirements, or even existing methodology descriptions (for the second scenario). Standard search methods based on syntactic techniques [5] are often inadequate to capture the similarity between documents, as they do not consider the semantics associated with the terms composing documents. For instance, without exploiting semantics, i.e., synonyms and related terms, the piece of document *D1* "*...**clients** for your small business enterprise...*" would wrongly be deemed as irrelevant to the query fragment *Q1 "....product requirements specified by the **customer**..."*. Moreover, terms might be *ambiguous*, i.e., they may have more than one possible meaning. For instance, even if the piece of document *D2* "*Distributed applications partition workloads between servers and **clients**...*" contains "client", the term is used in a completely different context, thus it should not be presented among the results.

In this paper, we propose a semantic method, implemented in the **Semantic Helper** component of the FACIT-SME solution, for searching ORM documents. It exploits a standard information retrieval weighting/ranking scheme extended to take into account synonyms and related terms information, together with Word Sense Disambiguation (WSD) techniques, and leads to the following achievements: (1) it is a fully automatic and semantic method that overcomes the standard syntactic technique limitations; (2) it is devised for IT SMEs, providing them with a flexible and easy-to-apply method that does not require big investments or knowledge prerequisites.

The rest of the paper is organized as follows: in Sections 2, we describe the Semantic Helper and the phases of the processes it supports; in Section 3, we describe the experimental evaluation of our method, while Section 4 concludes the work and briefly analyzes related works.

2 The Semantic Helper

The Semantic Helper supports the FACIT-SME solution by performing two main processes (see Figure 1):

1. **Semantic Glossary Population:** during this off-line process, statistical and semantic information are automatically extracted from the ORM documents and stored in a repository called Semantic Glossary;
2. **Relevant Document Ranking/Retrieval:** in this online process, user queries are processed and relevant documents are identified by exploiting the information provided by the Semantic Glossary.

In these two processes, we can identify three main phases: (a) keyword extraction and enrichment; (b) Semantic Glossary generation; (c) semantic similarity computation.

Keyword Extraction and Enrichment. The goal of this phase (involved in both processes) is to automatically extract, normalize and disambiguate terms

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

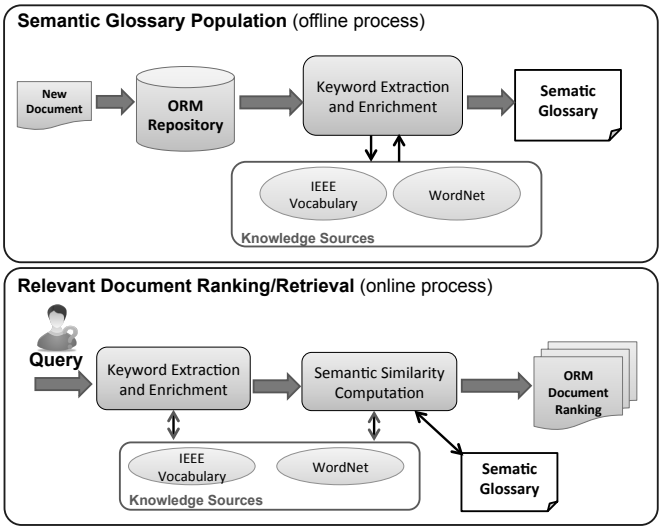

Fig. 1. Semantic Helper offline (top part) and online (bottom part) processes.

from the ORM documents collection/user queries. First of all, terms are extracted and normalized by means of *tokenization*, *stemming*, and *Part of Speech (POS) tagging*; moreover, possible *composite terms* (such as "product action plan" or "product requirements") are identified by means of a simple state machine and of POS tags information. Then, by exploiting external knowledge sources, the most relevant terms are selected. Finally, the selected terms[1] are annotated through a WSD algorithm, and associated with additional information such as definitions and synonyms.

The Semantic Helper makes use of two knowledge sources: the IEEE Software and Systems Engineering Vocabulary[2], and the WordNet[3] English thesaurus. These two sources differ in *Coverage*: the IEEE vocabulary includes only specialist terms in the project and computer science area, while WordNet complements it with general knowledge about English concepts; and *Granularity*: the IEEE vocabulary makes a very fine-grained sense distinction while WordNet makes coarser-grained sense distinctions. As in the FACIT-SME project we do not need a fine sense distinction (as errors may typically come out when terms have orthogonal senses), we employ WSD only for terms existing in WordNet (terms not existing in WordNet but in the IEEE vocabulary are associated with the first sense proposed).

To perform WSD, we use the STRIDER WSD algorithm described in [11]; however, other WSD algorithms might be employed as the ones described in [13, 14]. Given a document D containing a set of terms $\{t_1, t_2, ..., t_n\}$, for each term

[1] We limit the selection to nouns, as most of the semantics of a sentence is usually carried by noun terms [13].

[2] `http://www.computer.org/sevocab`

[3] `http://wordnet.princeton.edu/`

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

ANNOTATION	WN	IEEE	SYNS	DEFS	IDF	DOC_LIST
business enterprise#1	Y	-	commercial_enterprise#2, business#2	the activity of providing goods and services involving financial and commercial and industrial aspects.	0,6931	['D1']
client#3	Y	-	guest#4, node#7	any computer that is hooked up to a computer network; etc.	0,9315	['D2']
customer#1	Y	-	client#2	someone who pays for goods or services	0,8324	['D1']
…	…	…	…	…	…	…

Fig. 2. An excerpt of the Sense-Aware Semantic Glossary extracted from *D1* and *D2* (global view).

t_i, where $t_i \in D$, STRIDER returns an annotation, $A(t_i) = s_j$, where s_j is the top sense of a ranking $\{s_j, ..., s_k\}$ of plausible senses for t_i.

Semantic Glossary Generation. The Semantic Glossary is created the first time in the "offline" process, with all the documents available in the ORM; then it is automatically updated/enriched in case of new content is added to the ORM. It consists of a **global view** (all annotated terms in all documents, together with their statistics, see Figure 2), where each entry includes:

ANNOTATION: the annotation information in the form *"*$term_{\#senseIndex}$*"* (e.g., "$client_{\#3}$", where "client" has been annotated with the third sense proposed by the knowledge source);
WN and IEEE: if the annotation is w.r.t. WN or the IEEE vocabulary;
SYNS: possible synonym annotations;
DEFS: the definition corresponding to the annotation;
IDF: the annotation inverse document frequency [15] in the collection;
DOC_LIST: documents in which the annotation occurs.

and a **per-document view** (the annotated terms occurrences in each document with their statistics), containing:

DOC: the document ID in which the annotation occurs;
ANNOTATION: the annotation in the form *"*$term_{\#senseIndex}$*"*;
AF: the frequency of this annotation in the document, normalized by the total number of annotations in the document;
WEIGHT: the AF*IDF weight of the annotation.

Semantic Similarity Computation. The need of effectively and efficiently computing similarities between ORM/query documents is crucial to the project. To this end, we defined a *document similarity formula* $DSim(D^x, D^y)$ which, given a source document $D^x = \{t_1^x, ..., t_n^x\}$ (e.g., a given quality requirement description) and a target document $D^y = \{t_1^y, ..., t_n^y\}$ (e.g., each software methodology description available in the ORM), quantifies the similarity of the source document w.r.t. the target document. The computation of $DSim$ involves considering each annotation in D^x and finding the most similar annotation available in D^y by exploiting a *sense similarity formula* $SSim$. Thus, a **ranking** of the available documents is induced, predicting which documents are relevant and which are not w.r.t. D^x (i.e., the query document).

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Equation (1) shows the document similarity formula $DSim(D^x, D^y)$ which is given by the sum of all sense similarities $SSim$ between each annotation in D^x and the annotation (defined in (2)) in D^y maximizing the sense similarity with the annotated term in D^x:

$$DSim(D^x, D^y) = \sum_{t_i^x \in D^x} SSim(A(t_i^x), A(t^y_{\overrightarrow{j}(i)})) \cdot w_i^x \cdot w^y_{\overrightarrow{j}(i)} \quad (1)$$

$$A(t^y_{\overrightarrow{j}(i)}) = argmax_{t_j^y \in D^y}(SSim(A(t_i^x), A(t_j^y))) \quad (2)$$

where $A(t_i^x)$, $A(t_j^y)$ are the annotations of the i-th (j-th) term in the document x (y), respectively, and w_i^x and $w^y_{\overrightarrow{j}(i)}$ are the weights associated with the annotations and computed in the pre-document view of the Semantic Glossary. By exploiting weights common annotations, which are probably less meaningful, will give a lower contribute to the final similarity. $SSim$ is a sense similarity function which computes the similarity between two annotations:

$$SSim(A(t_i), A(t_j)) = \begin{cases} 1, & \text{if } A(t_i)\ SYN\ A(t_j) \\ r, & \text{if } A(t_i)\ REL\ A(t_j) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where the case of maximum similarity (value 1) corresponds to the case where the two annotations are synonyms (SYN relation). Moreover, the formula provides a further case where the two annotations are in some way related (REL relation) from a semantic point of view (i.e., broader/narrower sense etc.): such annotations will contribute with a similarity of r, where $0 < r < 1$. While the synonym information straightly comes from the Semantic Glossary, equation (4) shows a possible way of computing the similarity by exploiting the *glosses* (definitions) of the senses:

$$GSim(gl(A(t_i)), gl(A(t_j))) = \sum |ovl(gl(A(t_i)), gl(A(t_j)))|^2 \quad (4)$$

where $gl(A(t))$ is the gloss associated with the annotation of t. In this case, two annotations are semantically related if their *gloss similarity* $GSim$ (which quantifies the similarity by finding overlaps ovl between the two glosses [6]) exceeds a given threshold Th.

We also provide a further way of computing semantic relatedness by exploiting the relations between terms coming from the WordNet thesaurus [12]. Indeed, in WordNet senses (synsets) are connected to other synsets by hypernym (i.e., "is-a") relations. Two annotations (i.e., senses) are semantically related if their *hypernym similarity* $HSim$, which is defined as inversely proportional to the length of the shortest path connecting them (as in many knowledge management works [9, 11]), exceeds a given threshold Th.

Example. As simple summarizing example, let us consider again the pieces of documents $D1$ and $D2$ and the query fragment shown in Section 1. After the application of the keyword extraction and enrichment phase to $D1$ and $D2$, we obtain the Semantic Glossary (partially) shown in Figure 2. Thus, at query time, we need to apply WSD only to the terms extracted from the user query $Q1$. By computing the document similarity between $Q1$, $D1$ and $Q1$, $D2$, we obtain:

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

Query	Semantic			No WSD			Syntactic (no syn/rel) (baselines)		
	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
Q1	0,968	1,000	0,984	0,947	1,000	0,973	1,000	0,079	0,146
Q2	0,901	1,000	0,948	0,878	1,000	0,935	1,000	0,077	0,143
Q3	0,937	0,946	0,941	0,923	0,949	0,936	1,000	0,333	0,500
Q4	1	0,828	0,906	0,839	0,837	0,838	1,000	0,642	0,782
Q5	0,982	0,754	0,853	0,313	0,781	0,447	0,712	0,303	0,425
Q6	0,961	0,634	0,764	0,203	0,688	0,313	0,652	0,043	0,081

Fig. 3. Effectiveness analysis: precision, recall and F-measure (standard results for all-senses techniques on the left, two baselines on the right).

$$DSim(Q1, D1) = \sum_{t_i^{Q1} \in Q1} SSim(A(t_i^{Q1}), A(t_{\bar{j}(i)}^{D1})) \cdot w_i^{Q1} \cdot w_{\bar{j}(i)}^{D1} = 0.8$$

$$DSim(Q1, D2) = \sum_{t_i^{Q1} \in Q1} SSim(A(t_i^{Q1}), A(t_{\bar{j}(i)}^{D2})) \cdot w_i^{Q1} \cdot w_{\bar{j}(i)}^{D2} = 0$$

i.e., the document similarity between $Q1$ and $D2$ is equal to 0, and only the document $D1$ is returned as relevant for the query.

3 Experimental Evaluation

In this section, we present the results of the effectiveness evaluation we performed, on the proposed method, in the context of FACIT-SME. We formed a collection of 1500 documents (i.e., textual descriptions) about quality requirements taken from the ORM and derived from existing quality models, such as CMMI [2] and ISO 9000 [1]. Starting from this collection, we automatically generated the Semantic Glossary and we considered different queries simulating possible "From Scratch" scenario requests. Among them, we selected a set of 6 queries ($Q1 - Q6$) as the most representative ones: while queries $Q1 - Q3$ are mostly constituted by specific domain terms, queries $Q4 - Q6$ are representative of less common but possibly more complex to be handled requests, since they also contain several common and potentially more ambiguous terms.

For each query, we compared the output of the Semantic Helper with a "gold standard", i.e. relevant answers manually selected from the collection by experts in the field, and assessed precision, recall, and F-measure (Figure 3, left part). In the right part of the figure, we also present two baselines, i.e., the proposed semantic method working at term-level rather than at sense-level (no WSD applied, as in [12]), and a syntactic retrieval method ignoring synonyms and related terms, representative of document retrieval techniques commonly exploited by commercial systems. As we can see, the precision and recall levels achieved by the semantic method are generally very satisfying: all queries greatly benefit from semantic features such as synonyms and related terms management.We also notice that, due to the very specific and technical nature of the terms in $Q1 - Q3$, such queries are generally not largely affected by disambiguation issues.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

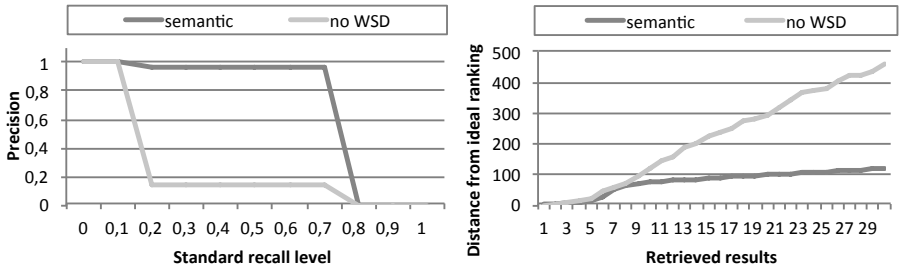

Fig. 4. In-depth effectiveness analysis for QT1: precision at standard recall levels (left) and distance from optimal ranking (right)

On the other hand, queries $Q4 - Q6$ contain several common and potentially ambiguous terms that are not present in the specialized IEEE vocabulary and for which only the use of WordNet similarity is allowed. This leads non-WSD techniques to several false-positives, especially in Q5 and Q6, while only the complete semantic method achieves satisfying precision/recall levels (at least more than double F measures in Figure 3). For instance, query $Q6$ contains, among others, the term "area": ORM contains few documents about "area" as "geographical region", while it contains several documents where "area" has the "subject of study" sense. Without disambiguation, we obtained several false positive documents corresponding to "subject of study" or containing falsely related terms, such as "topic", "issue" and "subject".

Finally, we deepened the effectiveness analysis by considering actual text documents (QT1-QT4) for which to find related documents in the collection, as in the "From Methodology" scenario. Such queries contain many terms and can possibly produce a very large number of results: thus, it is essential to evaluate also the induced ranking, so to assess whether the best suggestions are returned in the top positions. Figure 4 shows (on the left) the precision values obtained for QT1 (other queries performed very similarly) at different recall levels, i.e., when a given percentage of relevant documents have been found, and the distance from the ideal ranking (right). Our semantic method is compared to the non-WSD baseline: differently from the latter, it is able to filter the wrong senses and to identify the most significant terms, without being misled by non-relevant ones.

In conclusion, we can observe that, independently from the ambiguity of query terms, our semantic method leads to improvements in precision without compromising recall. Moreover, it gives also a key contribution in retrieving the most relevant results as first in the ranking.

4 Concluding Remarks

Several literature papers have highlighted possible benefits of combined knowledge engineering (KE) and software engineering methods for specific SE tasks [10, 7]. For instance, while standard reuse repositories are limited to plain syntactic search and, thus, generally suffer from low precision and recall [7], knowledge-based methods, such as [8] enhance the effectiveness of the component reuse task by proposing the usage of formal descriptions of components (in OWL) to

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,

be queried by specific graph query languages, such as SPARQL. Other notable proposals have been presented, for instance, for facilitating software process assessment through formal descriptions of specific improvement methods, such as CMMI [10]. As already noted in [7], however, the discussion on integrating SE and KE methods has been, in many cases, very academic, focusing on aspects like meta-modeling and neglecting applicability and usability.

Instead, the search method we presented in this paper is designed to be effective and easily applicable. Future work, in the upcoming project evaluation phase, will involve user IT companies in actual scenarios in order to obtain opinions about the effectiveness of the proposed techniques. Moreover, we will study how to adapt our search methodology in the agro-food domain in the context of the Biogest-Siteia projects [4], funded by Emilia-Romagna (Italy) regional government, which aims to increase the competitiveness of Regional seed companies through the use of modern selection technologies.

References

1. DIS 9001:2000 Quality Management Systems - Requirement. In *ISO TC176*, 1999.
2. Carnegie Mellon University Software Engineering Institute. In *CMMI for Development, Version 1.2*, 2006.
3. OSES Architecture and Component Specification. In G. Benguria, editor, *FP7-SME FACIT-SME (FP7-243695), Deliverable*, December 2010.
4. ORM Architecture and Engineering Models. In F. W. Jaekel, editor, *FP7-SME FACIT-SME (FP7-243695), Deliverable*, October 2010.
5. R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
6. S. Banerjee and T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *IJCAI*, pages 805–810, 2003.
7. H.-j. Happel and S. Seedorf. Applications of ontologies in software engineering. *Engineering*, pages 1–14, 2006.
8. C. Kiefer, A. Bernstein, and J. Tappolet. Mining Software Repositories with iSPAROL and a Software Evolution Ontology. In *MSR*, page 10. IEEE Computer Society, 2007.
9. C. Leacock and M. Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*, chapter 11, pages 265–283. The MIT Press, 1998.
10. H. Leung, L. Liao, and Y. Qu. A software process ontology and its application. In *the 4th International Semantic Web Conference*, 2005.
11. F. Mandreoli and R. Martoglia. Knowledge-based sense disambiguation (almost) for all structures. *Inf. Syst.*, 36(2):406–430, 2011.
12. R. Martoglia. Facilitate IT-Providing SMEs in Software Development: a Semantic Helper for Filtering and Searching Knowledge. In *SEKE*, pages 130–136. Knowledge Systems Institute Graduate School, 2011.
13. R. Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2), 2009.
14. L. Po and S. Sorrentino. Automatic generation of probabilistic relationships for improving schema matching. *Inf. Syst.*, 36(2):192–208, 2011.
15. G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[4] http://www.biogest-siteia.unimore.it/

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6,