Under consideration for publication in Knowledge and Information Systems

Exploiting Semantics for Filtering and Searching Knowledge in a Software Development Context

Sonia Bergamaschi[1], Riccardo Martoglia[2] and Serena Sorrentino[1]

[1]DIEF, University of Modena and Reggio Emilia, Modena, Italy
[2]FIM, University of Modena and Reggio Emilia, Modena, Italy

Abstract. Software development is still considered a bottleneck for SMEs (Small and Medium Enterprises) in the advance of the Information Society. Usually, SMEs store and collect a large number of software textual documentation; these documents might be profitably used to facilitate them in using (and re-using) Software Engineering methods for systematically designing their applications, thus reducing software development cost. Specific and semantics textual filtering/search mechanisms, supporting the identification of adequate processes and practices for the enterprise needs, are fundamental in this context. To this aim, we present an automatic document retrieval method based on semantic similarity and Word Sense Disambiguation (WSD) techniques. The proposal leverages on the strengths of both classic information retrieval and knowledge-based techniques, exploiting syntactical and semantic information provided by general and specific domain knowledge sources. For any SME, it is as easily and generally applicable as are the search techniques offered by common enterprise Content Management Systems (CMSs). Our method was developed within the FACIT-SME European FP-7 project, whose aim is to facilitate the diffusion of Software Engineering methods and best practices among SMEs. As shown by a detailed experimental evaluation, the achieved effectiveness goes well beyond typical retrieval solutions.

Keywords: knowledge management; text retrieval; semantic knowledge; semantic similarity; word sense disambiguation; software engineering.

1. Introduction and Motivations

One of the main bottlenecks for the development of the Information Society (*Aetic (Spain) and Agoria (Belgium) and AssInform (Italy) et. al.*, 24 October 2008)

Received xxx
Revised xxx
Accepted xxx

hes been software development, as the quality and productivity of work has not been able to keep up with the society software needs (*DG INFSO Internal Reflection Group on Software Technologies, ITEA*, April 2002; *Standish Group*, 2006). According to the analysis performed by the INNOSme project (InnoSME Project, 2008) across several countries, these issues are especially critical for software SMEs (Small and Medium Enterprises): the available resources cannot be devoted to new technology training as they are absorbed in the activity of software production. Thus, the integration between Software and Knowledge Engineering has become unavoidable in order to reduce time and cost for software development and to increase software quality (Happel and Seedorf, 2006; Binkley and Lawrie, 2010).

Usually, SMEs store and collect a large number of software textual documentation: indeed, text is everywhere and even test cases and inline comments could be useful knowledge sources (Lethbridge, Singer and Forward, 2003). This textual information might be profitably used to facilitate them in using (and re-using) Software Engineering methods for developing their applications; however, their inadequate information systems often prevents them from doing so (Garg, Goyal and Lather, 2010). To this aim, specific document filtering/search mechanisms based on textual similarity techniques are fundamental.

In literature, several methods for document filtering/searching in a software development context have been proposed (Happel and Seedorf, 2006), however a large number of fundamental challenges still need to be faced. Indeed, the great majority of these approaches is based on syntactic information retrieval techniques (Gyimothy, Ferenc and Siket, 2005; Poshyvanyk and Marcus, 2006)). Such approaches have found a wide application in the more general purpose Content Management Systems (CMSs), which are commercially available and can be easily adopted and exploited by SMEs (Varghese and Systems, 2012).

Standard syntactical search techniques (Baeza-Yates and Ribeiro-Neto, 1999)[1] often suffer of low effectiveness as they are inadequate to capture the similarity between documents and disregard the semantic connections (*synonyms or semantic relations*) of the terms composing them. For instance, let us consider the piece of document "*...**clients** for your activity...*" and the fragment of query *"...product requirements specified by the **customer**..."* (e.g., from the Methodology scenario). In a syntactical search approach, no match would be found between the query and the document, as they have no words in common. On the contrary, by adding semantics, i.e., synonyms and related terms (i.e., broader, narrower or correlated terms), it is possible to discover that "customer" is a synonym of "client" and, thus, that this document may be relevant for the query.

Other approaches make also use of semantic methods based on dictionary or thesauri (e.g.,WordNet[2]) by considering synonyms and related terms (Shepherd, Fry, Hill, Pollock and Vijay-Shanker, 2007; Sridhara, Hill, Pollock and Vijay-Shanker, 2008). However, they do not consider the term *ambiguity* problem which may affect the effectiveness of the method: a term may have more than one possible meaning (e.g., "client" means "someone who pay for goods or services" when used in a Business context, while it means "any computer that is hooked up to a computer network" when used in a Computer Science context). For instance, let us consider the piece of document "***Clients** and servers exchange*

[1] Roughly speaking, these techniques look for documents containing the same terms specified by the user query.
[2] `http://wordnet.princeton.edu/`

messages in a request-response messaging pattern...". As regards the previous query, this document is potentially relevant, since it contains "client" which is, in its Business sense, a synonym of "customer". However, in this piece of document "client" is used in a Computer Science context, thus it is not applicable to the query.

Finally, other more complex approaches require the knowledge of technical languages (such as SPARQL) to be used (Happel, Korthaus, Seedorf and Tomczyk, 2006; Kiefer, Bernstein and Tappolet, 2007; Leung, Liao and Qu, 2005; Soydan, 2006; Witte, Zhang and Rilling, 2007) and thus, they are not suitable for non skilled users in SMEs.

Starting from these considerations, in this paper we propose a fully automatic and semantic approach for filtering/searching software documentation, carefully considering the actual user-targets. The proposed solution significantly extends the preliminary work presented in (Martoglia, 2011) and, building on some of the initial ideas anticipated in the short communication (Bergamaschi, Martoglia and Sorrentino, 2012), it aims to:

1. be *easily and generally applicable/maintainable* by IT SMEs: it allows users to look for information by specifying simple keyword queries or document queries, i.e. by simply submitting existing documents to the system; moreover, it does not require big investments or knowledge prerequisites: it exploits the large amounts of textual documents (i.e., methodology descriptions, and so on) already available in each enterprise, without requiring any conversion towards complex structured formats which would be time and cost consuming;

2. *effectively and automatically* identify the similarities between such queries and a reference set of documents. The limitations of standard syntactical techniques (such as the ones usually exploited by enterprise CMSs) are overcome by considering the *semantics* intrinsically associated to the document/query terms and by addressing the problem of term ambiguity through the use of Word Sense Disambiguation (WSD). To this aim, we exploit different kinds of external knowledge sources (both general and specific domain dictionaries or thesauri);

3. be *flexible* so to become a basis of many high-level functionalities, i.e., filtering software methodologies for software process assessment and improvement, quality requirements for helping in certification process, best practices for facilitating knowledge sharing, and so on.

Our method has been studied and evaluated in the context of the European FP7 3-years project "Facilitate IT-providing SMEs by Operation-related Models and Methods" (FACIT-SME Project, 2010-2012), and is implemented in the *Semantic Helper* component of the FACIT-SME solution.

In the rest of the paper, Section 2 gives an overview of the overall FACIT-SME scenario and of the proposed semantic helper. In Sections 3 and 4, we focus on the analysis and semantic techniques on which it is based. Section 5 analyzes the related work, while the detailed experimental evaluation presented in Section 6 shows the achieved effectiveness results, going beyond typical retrieval solutions. Finally, Section 7 concludes the work.

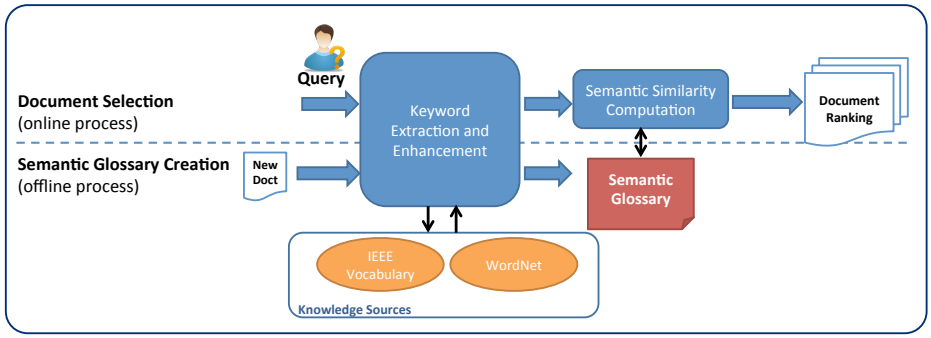

Fig. 1. A schematic overview of the Semantic Helper processes.

2. FACIT-SME Solution and Semantic Helper Overview

The target of the FACIT-SME project is to promote the use of SE methods within IT SMEs, for designing and developing their applications integrated with the business processes in a more systematic way. Another goal is to provide efficient and affordable certification of these processes according to internationally accepted standards, and to securely share best practices, tools and experiences with development partners and customers. The achievements FACIT-SME are the following: a novel Open Reference Model (ORM) (*ORM Architecture and Engineering Models*, October 2010) for ICT SMEs serving as an underlying knowledge Backbone; On top of the ORM, a customizable Open Source Enactment System (OSES)) (*OSES Architecture and Component Specification*, December 2010) as IT support for the project-specific application of the ORM. More specifically, the ORM 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, specific search mechanisms, representing a key part of the OSES, support the identification of adequate processes and data structures for a specific enterprise. The inputs of the search mechanism are the company and project information and/or the existing methodology descriptions. Then, through a filtering phase, the organization receives a set of suggestions in the form of the most relevant/useful elements and models in the ORM. Besides five R&D partners providing the required competencies, the project consortium also includes five SMEs operating in the IT domain which will evaluate the results in daily-life applications.

The **Semantic Helper** is the FACIT-SME software component which implements our method. The goal of the Semantic Helper is to filter and search the relevant information available in the ORM, in two possible scenarios:

- **"from scratch" scenario:** assisted filtering/selection of ORM elements given specific enterprise objectives, e.g., to give pointers to useful information for helping the company in achieving its certification objectives;
- **"from methodology" scenario:** assisted suggestion proposal for a given enterprise methodology, e.g., to help identifying relevant information or gaps between the given methodology and the ORM methodologies.

To this end, a representation of the key parts of the ORM in a semantic and

machine-processable way is needed. The Semantic Helper supports two main processes to deal with textual information (see Figure 1):

1. **Semantic Glossary Creation:** this is an off-line process where the Semantic Helper automatically extracts and "normalizes" a shared "terminology" from the given set of ORM documents (i.e., quality requirements or software methodologies), eventually populating the computer-processable Semantic Glossary; such information will be used during the online (document retrieval) process. In particular, the extracted terminology is also enhanced with statistical and semantic information (i.e., links to thesauri and domain vocabularies, definitions, and synonyms);
2. **Document Selection:** : it is an online process where user queries are processed and, consequently, the relevant documents are selected. First of all, the query document (e.g., existing enterprise documentation) is analyzed by means of the same techniques used for the Semantic Glossary creation. Once the query has been reduced to a set of terms with associated semantics, appropriate semantic similarity techniques are exploited to easily identify relevant ORM documents, and to produce a list of suggestions ranked on the similarity (relevance) score.

The keyword extraction and enhancement phase, involved in both processes, is detailed in Section 3.1; in Section 3.2, the structure of the semantic glossary produced in the offline process is described, while the semantic similarity computation phase, involved in the online process, is detailed in Section 3.3.

3. Semantic Helper Techniques

3.1. Keyword Extraction and Enhancement

Our goal for keyword extraction and enhancement has been to design and develop an effective and easy-to-apply technique for automatically analyzing text and extracting terms, together with their associated semantics and statistics. In particular, we wanted to devise a flexible technique to be exploited both for "off-line" analysis (thus working on the textual descriptions already available in the ORM) and for "on-line" querying operations, i.e., applied on the fly to the submitted query documents. The keyword extraction and enhancement phase is composed by the following steps:

1. **Tokenization:** terms are identified and punctuation is removed;
2. **Stemming:** the tokens are "normalized" and "stemmed", i.e., terms are reduced to their base form (managing plurals and inflections);
3. **POS (Part of Speech) Tagging:** the tokens are "tagged" with Part of Speech tags (i.e., nouns, verbs, ...);
4. **Composite term identification:** 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;
5. **Filtering and enhancement:** by exploiting external knowledge sources, the most relevant terms are selected and associated to additional information (such as definitions and synonyms). More specifically, we made use of the IEEE

TERM	WN	IEEE	SYNS	DEFS	IDF	DOC_LIST
acquirer	Y	Y	buyer, customer, owner, purchaser	(1) stakeholder that acquires or procures a product or service from a …	7.4961	['QM1372']
acquisition	Y	Y	outsourcing	(1) process of obtaining a system, software product or software service …	5.5491	['QM0392', 'QM0755', …]

Fig. 2. An excerpt of the FACIT-SME Semantic Glossary (global view).

Software and Systems Engineering Vocabulary[3], a knowledge source covering specialist terms in the project area, and the WordNet English thesaurus (Miller, 1994), complementing the specialistic source with general knowledge about English concepts;

6. **Term statistics and weight computation:** weights are computed for each term, reflecting their relevance and meaningfulness in the document.

Even if our techniques are able to extract terms belonging to different parts of speech, we limit the extraction to nouns, as most of the semantics of a sentence is usually carried by noun terms (Navigli, 2009).

3.2. The Semantic Glossary

By applying batch keyword extraction and enhancement to the documents currently available in the ORM, we achieved a first significant result in the FACIT-SME project, i.e., the automatic generation of the **Semantic Glossary**. This first draft can be automatically updated/enriched whenever new content is added to the ORM, while more fine-grained user interventions for adding/modifying/eliminating information are also possible.

The Semantic Glossary stores all the terms in all the documents with their statistics (**global view**) and the terms occurrences in each document with their statistics (**per-document view**). The glossary global view is an alphabetical sort of all the extracted terms, in a tabular form. Figure 2 shows an excerpt of it. The format is:

TERM: the extracted term;

WN: whether it is present in the WordNet thesaurus;

IEEE: whether it is present in the IEEE vocabulary;

SYNS: possible synonyms for the term (as extracted from the IEEE vocabulary and/or WordNet);

DEFS: possible definitions for the term (as extracted from the IEEE vocabulary and/or WordNet);

IDF: the inverse document frequency of the term in the collection;

DOC_LIST: a list of the documents IDs in which the term occurs.

Note that, with "possible" synonyms and definitions, we mean the collection of synonyms and definitions available in the knowledge sources for the different meanings associated to the terms. The glossary **per-document view** is a list of

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

DOC	TERM	TF	WEIGHT(TF*IDF)
QM0001	iso	1	6.8024
QM0002	management	0.3333	0.6931
QM0002	quality	0.3333	1.0303

Fig. 3. An excerpt of the FACIT-SME Semantic Glossary (per-doc view).

all the term occurrences in the documents, sorted by the document ID, together with their statistics (see an excerpt in Figure 3). For each term in each document, the view contains:

DOC: ID of the document containing the term;

TERM: the term extracted;

TF: the frequency of this term in the document, normalized by the total number of terms in the document;

WEIGHT: the TF*IDF weight of the term.

As we will see in the next section, the content of the glossary allows the similarity functions of the Semantic Helper to draw useful knowledge from both the semantic and the text retrieval research areas. Moreover, by exploiting the weight TF*IDF[4] (Salton and Buckley, 1988) in the similarity computation, common terms, which are probably less meaningful, will give a lower contribute to the final similarity (since they will have a low weight).

The similarity techniques described in the next section are designed to work automatically without further intervention (and, as proved in Section 6, provide encouraging results). Nevertheless, the list of synonyms and definitions retrieved from the external knowledge sources can be automatically refined by means of WSD techniques (as we will deeply investigate in Section 4) in order to maximize retrieval effectiveness.

3.3. Semantic Similarity Computation

As anticipated in the past sections, the need of effectively and efficiently computing similarities between documents is crucial in contexts like the one of the project. With this goal in mind, we aim to define a *document similarity formula* $DSim(D^x, D^y)$: for a given a source document $D^x = \{t_1^x, ..., t_n^x\}$ and a target document $D^y = \{t_1^y, ..., t_n^y\}$, the formula expresses the similarity of the source document w.r.t. the target document. In particular, the computation of $DSim$ between a given D^x (e.g., a given quality requirement description) and each possible submitted D^y (i.e., each available software methodology description of the ORM) involves the following steps:

1. considering each term in D^x and finding the most similar term or terms available in D^y by exploiting a *term similarity formula* $TSim$. Computing $TSim$ means to identify:
   (a) equal terms;

[4] IDF is obtained by dividing the total number of documents by the number of documents containing the term and then by computing the logarithm of that ratio.

(b) synonym terms (if no equal terms are found);
(c) semantically related terms (if no synonym terms are found).

2. inducing a **ranking** of the available documents (on the basis of $DSim$), thus predicting which documents are relevant and which are not w.r.t. D^x.

Let us now discuss in detail the proposed formulas for $DSim$ and $TSim$. The document similarity formula between a given source document D^x and a target document D^y is shown in Equation (1): the similarity is given by the sum (defined in (2)) of all term similarities between each term in D^x and each term in D^y maximizing the term similarity with the term in D^x:

$$DSim(D^i, D^j) = \sum_{t_i^x \in D^x} TSim(t_i^x, t^y_{\bar{j}(i)}) \cdot w_i^x \cdot w^y_{\bar{j}(i)} \tag{1}$$

$$t^y_{\bar{j}(i)} = argmax_{t_j^y \in D^y}(TSim(t_i^x, t_j^y)) \tag{2}$$

where $w_i^x = tf_i^x \cdot idf_i$ and $w^y_{\bar{j}(i)} = tf^y_{\bar{j}(i)} \cdot idf_{\bar{j}(i)}$. In this way, each term contributes to the final similarity with a different weight, i.e., more frequent and more significant terms contribute more to the similarity between the two documents[5]. $TSim$ is computed by means of Equation (3) which considers **synonyms** (thus, implicitly equal terms) and **semantically related terms**:

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

Note that the case of maximum similarity (i.e. value 1) holds when the two terms are synonyms (SYN relation). Moreover, when the two terms are in some way related from a semantic point of view (i.e., broader/narrower terms etc.), the formula provides a similarity value of r, where $0 < r < 1$ is a user-defined fixed similarity value.

Besides synonym information, we consider two different ways to determine whether two terms are related. Equation (5) shows a possible way of computing the similarity by exploiting the *glosses* (definitions) of the terms:

$$t_i \ REL \ t_j \iff GSim(gl(t_i), gl(t_j)) \geq Th \tag{4}$$

$$GSim(gl(t_i), gl(t_j)) = \sum | \ ovl(gl(t_i), gl(t_j)) \ |^2 \tag{5}$$

Two terms are in relation REL, thus semantically related, if their gloss similarity, $GSim$ exceeds a given threshold Th. The Literature presents many possible ways of computing similarities between glosses (Navigli, 2009). We decided to use the extended gloss overlap measure (Banerjee and Pedersen, 2003) shown in (7), being it one of the most popular and effective. It quantifies the similarity between the two glosses by finding overlaps in them (the similarity is the sum of the squares of the overlap lengths). If one or both terms are associated to more than one gloss, the formula returns the similarity between the two closest glosses.

[5] Note that, Equation (1) is not meant to be symmetric, instead it is conceived so to facilitate the ranking of documents D^y w.r.t. document D^x. In case symmetry is needed, the summation in (1) can be extended to the terms of both documents.

Exploiting the relations between terms coming from the WordNet thesaurus is another possibility to compute semantic relatedness. Indeed, in WordNet terms are associated to one or more different meanings (or senses), and each term is then is connected to other terms meanings by hypernym (i.e., "is-a") relations[6] (Miller, 1994). We adopt one of the most widely used methods in knowledge management, relying on the hypernym relations:

$$t_i RELt_j \iff HSim(t_i, t_j) \geq Th \tag{6}$$

$$HSim(t_i, t_j) = \begin{cases} -ln\frac{len(t_i,t_j)}{2H}, & \text{if } \exists\ lca(t_i, t_j) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In our case, two terms are semantically related if their hypernym similarity $HSim$ exceeds a given threshold Th. In particular, the $HSim$ shown in (7) derives from the works (Mandreoli and Martoglia, 2011; Leacock and Chodorow, 1998) and computes a score which is inversely proportional to the length of the shortest path connecting the (senses of the) two terms. H is a constant representing the maximum depth of the hypernym tree, which for WordNet is defined as 16. On the other hand, the similarity is 0 if the two terms are not connected in the WordNet hypernym structure.

Let us now see how the keyword extraction/enhancement and semantic techniques, we just presented, can be used in the context of a small illustrative example.

Example 1. Let us suppose that $D1$ is a fragment of a document available in the ORM repository:

D1. "*How to get more **clients** for your small business **enterprise***"

Given the following queries:

*Q1. "product requirements specified by the **customer**"*
*Q2. "our **organization**'s design takes a hierarchical structure"*

$D1$ has no terms in common with both $Q1$ and $Q2$. However, by using the Semantic Helper techniques, we can easily determine that $D1$ might contain information potentially relevant to $Q1$, as "customer" is a synonym of "client". Further, by analyzing the semantic similarity of the terms in $Q2$ w.r.t. those already available in the Semantic Glossary, the term "organization" can easily be found as strictly related to "enterprise" by means of formulas (5) or (7). Therefore, $D1$ will also be presented as a possible "suggestion" for $Q2$, even if, typically, with a lower score (related terms usually contribute to weaker scores than synonyms or equal terms). ⋄

4. Beyond Term Ambiguities: Sense-Aware Techniques

Human Language is intrinsically ambiguous, and terms may be *polysemous*, i.e., they may have different senses (or meanings) on the basis of the context where

[6] We recall that t_i is said to be a hypernym of t_j if there exists a t_i's meaning that includes (i.e., is a hypernym) of a meaning of t_j: for instance, "electronic device" is a hypernym of "computer".

they are used. For example, as we have previously seen in Section 1, the term "client" may be employed in a Business context with the meaning of "customer" or in a Computer Science context with the meaning of "computer". Therefore, when user queries and/or documents contain polysemous terms, the similarity techniques described so far may not be sufficient to compute document similarity, and several non-relevant documents might be returned. Let us consider two motivating examples.

Example 2. Starting from the scenario illustrated in Example 1, let us suppose that the ORM contains also document $D2$:

D2. "*Distributed applications partition workloads between the servers and* ***clients***"

As regards the query $Q1$, both documents are potentially relevant, since they both contain the term "client" which is, in its Business sense, a synonym of "customer". However, while in $D1$ "client" is used in its Business sense, in $D2$ it is used in a Computer Science context: in this case, only $D1$ is pertaining to the user query, while $D2$ should be discarded. ⋄

Example 3. Let us now consider the following additional query:

Q3. "*manage the risk of* ***computer*** *breakdown to avoid losing information*"

This query has no terms in common with the documents. However, by analyzing their semantic similarity, we can find out that the term "computer" in $Q3$ is strictly related to the term "client" by means of formula (7), as "computer" is a hypernym of "client" in WordNet. However, this is true only when "client" is used in a Computer Science context. As a consequence, only document $D2$ is relevant to the query. ⋄

To address term ambiguity problems in the FACIT-SME context, the idea is to exploit Word Sense Disambiguation (WSD), i.e., a Natural Language Processing technique for automatically (or semi-automatically) identifying the sense of a term in a context (Navigli, 2009). Indeed, term senses represent strategic information in order to avoid such "pitfalls" as the ones illustrated in the above examples. Specifically, by using WSD, we can improve the searching/filtering mechanisms in two ways: (1) by excluding documents containing *false-synonyms* of the query keywords (such as terms "client" and "customer" in Example 2); (2) by excluding documents containing *false-related terms* (as discussed in Example 3 for document $D1$). As further examples of the first aspect, in WordNet both terms "customer" and "node" are potential synonyms of "client"; however, when "client" is used in a Business context, only "customer" is a true-synonym, while "node" is a false-synonym. As to the second aspect, the term "client" is related to the term "website" only when it is used in a Computer Science context, while when it is in a Business context, "website" represents a false-related term.

Starting from these considerations, we decided to enhance the Similarity Techniques employed in the FACIT-SME Semantic Helper, by including the sense information deriving from the application of WSD techniques. In the following sections, we describe the extensions needed to make the Semantic Glossary and the similarity formulas aware of the senses of terms in their context[7].

[7] In the following, we will denote the new sense-discerning techniques as "sense-aware", while the original ones described in Section 3 will be denoted as "all-senses".

4.1. Sense-Aware Extensions to the Keyword Extraction and Enhancement

WSD is usually applied on text and it is performed w.r.t. a knowledge source (e.g., glossaries, thesauri or dictionaries) representing the sense inventory (i.e., all the possible meanings of terms) (Navigli, 2009). As previously described, the Semantic Helper makes use of two knowledge sources: the IEEE vocabulary and WordNet. The main differences between these two sources are in:

1. *Coverage:* the IEEE vocabulary includes only terms and senses belonging to the Computer Science domain, and thus, it does not contain common terms like "people" and "work", which instead may be present in queries and documents; on the contrary, WordNet covers a wide range of domains but it misses several specific terms belonging to the FACIT project domain (e.g., "regression test" or "testability");
2. *Granularity:* the IEEE vocabulary makes such a fine-grained sense distinction that for many users it is often difficult to distinguish between senses (e.g., for "architecture" two distinct senses exist for "fundamental organization of a system embodied in its components" and "organizational structure of a system and its implementation guidelines"); on the contrary WordNet makes coarser-grained sense distinctions (for instance, distinguishing between "profession of designing buildings" and "organization of a computer's hardware or system software").

As described in (Palmer, Dang and Fellbaum, 2007), the performance of WSD strictly depends on the granularity of the sense distinctions, which should be selected on the basis of the application. In a software development context, we do not need a fine sense distinction; instead, errors may typically come out when terms have orthogonal senses (as in the case of "client"). Starting from these considerations, we decided that we needed WSD only for terms existing in WordNet. If a term does not exist in WordNet, we check if it is present in the IEEE vocabulary, and in this case, we associate the first sense proposed to it.

We used the STRIDER WSD algorithm described in in (Mandreoli and Martoglia, 2011; Mandreoli, Martoglia and Ronchetti, 2005) to perform WSD; other WSD algorithms might be employed as the ones described in (Po and Sorrentino, 2011), developed in the context of the MOMIS Data Integration System (Beneventano, Bergamaschi, Guerra and Vincini, 2001), and in (Navigli, 2009). STRIDER is designed to perform effective disambiguation of terms w.r.t. the WordNet thesaurus, also managing structures that go beyond plain text (e.g., nodes in XML trees or RDF graphs). Its outcome is a ranking of the plausible senses for each term, from which the top sense is automatically suggested.

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 each term t_i occurring in a given document D. We apply WSD to all the documents available in the ORM: WSD becomes a new step to be performed during the keyword extraction and enhancement phase (see Section 3.1). Further, the Semantic Glossary structure needs to be extended for storing the WSD information: we need to associate each term to the corresponding annotation and the list of documents where the term is used with that sense, thus going toward a sense-aware semantic glossary.

We modified the previous Semantic Glossary structure in the following way: all references to terms (also including synonyms, frequency information, and so on) become references to senses. Figure 4 shows an excerpt of the sense-

ANNOTATION	WN	IEEE	SYNS	DEFS	IDF	DOC_LIST
bank#1	Y	-	banking_concern#1, depository_financial_institution#1, banking company#1	a financial institution that accepts deposits and channels the money into lending activities.	0.6931	[D1]
business enterprise#1	Y	-	commercial_enterprise#2, business#2	the activity of providing goods and services involving financial and commercial and industrial aspects.	1.795	[D2]
client#3	Y	-	guest#4, node#7	any computer that is hooked up to a computer network; etc.	1.624	[D1]
computer#1	Y	-	computing_machine#1, computing_device#1, data_processor#1, electronic_computer#1, information_processing_ system#1	a machine for performing calculations automatically	1.538	[D1]
customer#1	Y	-	client#2	someone who pays for goods or services	0.952	[D2]
...	...	...	...	…	...	...

Fig. 4. An excerpt of the Sense-Aware Semantic Glossary extracted from *D1* and *D2* (global view).

aware Semantic Glossary (global view), as extracted from documents $D1$ and $D2$ of our previous examples. As we can see, the TERM column becomes the *ANNOTATION* column, containing the annotation information in the form *"*$term_{\#senseIndex}$*"* (e.g., "$client_{\#3}$" means that the term "client" has been annotated with the third sense proposed by the knowledge source). Moreover, TF, i.e., term frequency, becomes AF (annotation frequency), i.e., the frequency of a specific term annotation.

The other columns do not change their name but have some significant changes in their content: WN and IEEE inform whether the annotation is w.r.t. WN or w.r.t. the IEEE vocabulary; SYNS contains the synonym annotations (e.g., for the ANNOTATION "$client_{\#3}$", it contains "$guest_{\#4}$" as the fourth meaning of "guest" is a synonym of the third sense of client); DEFS contains only the definition corresponding to the annotation (e.g., for "$client_{\#3}$", "any computer that is hooked up to a computer network"); DOC_LIST contains the list of documents in which the term occurs with the same annotation; finally, IDF and WEIGHT are computed on the basis of the annotation frequency AF.

4.2. Sense-Aware Semantic Similarity

We now need to enhance the document similarity techniques in order to fully exploit the new information available in the sense-aware Semantic Glossary. In particular, the formula $DSim(D^x, D^y)$ has to be modified; we define the revised $DSim$ as:

$$DSim(D^x, D^y) = \sum_{t_i^x \in D^x} SSim(A(t_i^x), A(t^y_{\bar{j}(i)})) \cdot w_i^x \cdot w^y_{\bar{j}(i)} \tag{8}$$

$$A(t^y_{\bar{j}(i)}) = argmax_{t_j^y \in D^y}(SSim(A(t_i^x), A(t_j^y))) \tag{9}$$

where $A(t_i^x)$ is the annotation of the i-th term in document x (i.e., the sense associated to term t_i in document x), and $A(t_j^y)$ is the annotation of the j-th term in document y (i.e., the sense associated to term t_j in document y). $SSim$ is a sense similarity function which computes the similarity between two annotations, i.e., between the senses associated to two terms. It is defined as follows:

$$SSim(A(t_i), A(t_j)) = \begin{cases} 1, & \text{if } A(t_i)\ SYN\ A(t_j) \\ r, & \text{if } A(t_i)\ REL\ A(t_j) \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

where $A(t_i)\ SYN\ (t_j)$ means that the two annotations are the same. The REL relation can be computed, as always, through the functions $HSim$ and $GSim$; the only difference in this case is that, instead of considering all the possible senses for a term, we restrict the computation to the senses specified in the annotations. Thus, the sense-aware formulas for $GSim$ and $HSim$ will be:

$$GSim(gl(A(t_i)), gl(a(t_j))) = \sum \mid ovl(gl(A(t_i)), gl(A(t_j))) \mid^2 \quad (11)$$

$$HSim(A(t_i), A(t_j)) = \begin{cases} -ln\frac{len(A(t_i), A(t_j))}{2H}, & \text{if } \exists\ lca(A(t_i), A(t_j)) \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

4.3. Sense-Aware Techniques in Practice

When a user submits a query, the Semantic Glossary already contains the term document annotations $A(t_j^y) \in D^y$. Therefore, at run time, we need to apply our WSD algorithm only to the user query's terms.

Notice that there exist cases where it is not convenient to apply WSD to the query terms. Let us consider, for instance, a possible keyword query "guest address": in this case, if no other keyword is available, there is not enough context information to determine if "guest" is a computer in the network or a visitor, as well as if "address" means a computer address or a street address. As a consequence, in this and in other similar cases, we consider all the possible senses for the query terms, since WSD would not have sufficient information to perform annotations. Further, we still apply formulas (8, 9, 10) by computing $SSim$ for each possible annotation of the query terms (and, then, by considering the maximum value). We conclude the section by providing an example of how the sense-aware similarity techniques actually work on a simple case.

Example 4. Let us consider query $Q1$ and documents $D1$ and $D2$ of our previous examples. As previously seen, for this query only document $D1$ should be returned by the semantic helper, while document $D2$ should be discharged. Let us see how this is accomplished by our sense-aware techniques.

The Semantic Glossary shown in Figure 4 is obtained after the application of the Semantic Glossary process described in subsections 3.1 and 3.2. The composite term identification step identifies the composite term "business enterprise" and "client request". Neither WordNet nor the IEEE vocabulary provide an entry for "client request": in these cases, we consider and disambiguate the single terms (i.e., "client" and "request")[8]. Then, we automatically disambiguate the query terms. The returned query annotations are: $client_{\#3}$, $server_{\#3}$, and

[8] Other approaches to deal with composite terms, as the one described in (Sorrentino, Bergamaschi and Gawinecki, 2011), could be employed.

$communication_{\#1}$. Now, we can compute the document similarity between $Q1$, $D1$ and $Q1$, $D2$[9]:

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

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

i.e., the similarity between $Q1$ and $D2$ is null, and only $D1$ is selected as relevant for the query.$\diamond$

5. Related Work

In literature, several approaches applying *syntactic information retrieval techniques* to specific software engineering tasks have been proposed. For instance, the well-established notions of vector space model, tf-idf weighting and preprocessing techniques for stemming and stopword removal are exploited in a number of works (e.g., (Gyimothy et al., 2005; Poshyvanyk and Marcus, 2006; Trakarnviroj and Prompoon, 2012)), as also noted in a recent survey on sofware maintenance and evolution (Binkley and Lawrie, 2010). Generally speaking, such methods are focused on very specific tasks or scenarios (and, therefore, specific kinds of input information). Their results confirm the possible applicability of standard information retrieval techniques (such as the ones from which we started to devise our approach) to SE scenarios, even if they are limited by the absence of semantic analysis.

A "classic" information retrieval foundation also characterizes the more general purpose CMSs which are commercially available and can be easily adopted and exploited by SMEs (Varghese and Systems, 2012) (a notable example is Alfresco[10]). Similarly to our approach, the automatic nature, general applicability and ease of use of these CMSs makes it easy to search for information also for non skilled users, for instance by allowing them to compose queries in a simple keyword-based way. However, in such CMSs only syntactic features are exploited, while the semantics of terms is not taken into account, therefore limiting the achievable results (see also Section 6 for an experimental comparison with our technique).

Several papers tried to go beyond syntactic retrieval techniques, showing the possible benefits of exploiting *semantic knowledge-based methods* for specific SE tasks (Gall, Lukins, Etzkorn, Gholston, Farrington, Utley, Fortune and Virani, 2008; Girardi and Ibrahim, 1994; Shepherd et al., 2007; Sridhara et al., 2008; Udomchaiporn, Prompoon and Kanongchaiyos, 2006). In fact, standard reuse repositories are limited to plain syntactical search and generally suffer from low effectiveness, as it was also stated in (Happel and Seedorf, 2006). On the other hand, knowledge-based approaches can really enhance the effectiveness of the component reuse task by proposing the usage of semantics. In (Girardi and Ibrahim, 1994) a software reuse system based on the processing of the natural language descriptions of software artifacts is described. The retrieval mechanism

[9] In this example, we set for $GSim$ a default threshold of 10 and for $HSim$ a default threshold of 0.25.

[10] `http://www.alfresco.com/`). Other commercial tools offer analogous functionalities.

is based on a similarity analysis which exploits synonym, hypernym and hyponym relationships extracted from WordNet. In (Shepherd et al., 2007) the authors propose a tool called FindConcept able to expand search queries with synonyms from WordNet. Furthermore, other notable works propose and present methods for retrieving information from specifically structured software documents or artifacts. For instance, in (Girardi and Ibrahim, 1994; Sridhara et al., 2008) the retrieving process is performed by focusing on the comments that can be found all through the software code. The approach (Gall et al., 2008) is based on semantic metrics, calculated by first extracting class names and the relevant paragraps from IEEE-formatted design documents.

Generally speaking, most of the available knowledge-based SE methods cannot be directly compared to ours, since they offer solutions tailored for very specific SE tasks and/or very specifically structured documents, rather than a general method that can be applied on any kind of textual documentation without any preparation or prerequisite. Further, most available approaches limit their semantic features to the use of synonyms and related terms extracted from general lexical resources such as WordNet (Shepherd et al., 2007; Sridhara et al., 2008). Instead, as WordNet does not contain several software engineering terms, we chose to also exploit specific domain vocabularies for the Software Engineering context. Further, we take advantage not only of synonyms and related terms but also of WSD techniques able to capture the meaning of terms in a context.

Finally, some approaches go even beyond the above mentioned semantic techniques by exploiting formal descriptions and representations of the software information (Constantopoulos, Jarke, Mylopoulos and Vassiliou, 1995; Happel et al., 2006; Kiefer et al., 2007; Leung et al., 2005; Mylopoulos, Borgida, Jarke and Koubarakis, 1990; Soydan, 2006; Witte et al., 2007). For instance, (Mylopoulos et al., 1990) exploits the popular Telos language to represent requirements, design and code; other approaches make use of ontologies to describe the functionality of components (Happel et al., 2006; Kiefer et al., 2007; Leung et al., 2005; Soydan, 2006). The use of such knowledge representation formalisms allows convenient and powerful querying, for instance by using SPARQL (Happel et al., 2006). However, this kind of approaches requires expert knowledge management skills in order to create the queries. Moreover, the use of specialized ontologies requires their complete update when new sources (e.g. new documents) are added to the repository, making these solutions hardly suitable or even applicable in the context of small enterprises. The discussion on integrating SE and KE approaches has been, in many cases, too academic, often neglecting the applicability and usability issues, as observed in (Happel and Seedorf, 2006). On the contrary, our method allows anyone to create queries by simply using keywords (or already existing pieces of documentation), while offering at the same time the benefits given by advanced semantic document management.

6. Experimental Evaluation

In this section, we present the results of the evaluation of our tool within the FACIT-SME project. We collected a set of 1500 documents in the domain of quality requirements, containing approximately 25000 words in total, taken from project partners and from well-known quality models, such as CMMI (*Carnegie Mellon University Software Engineering Institute*, 2006) and ISO 9000 (*DIS 9001:2000 Quality Management Systems - Requirement (pdf)*, 1999). Starting

Query Number	Main keywords extracted from the queries
Q1	ISO, Interface, requirement,...
Q2	configuration management system, project management,...
Q3	supplier, traceability, identifier,...
Q4	methodology, purpose, recovery,...
Q5	audit, environment, brainstorming,...
Q6	hardware, software, documentation,...
Q7	market, provider, work,...
Q8	scheme, organization, skill,...
Q9	area, meeting, plan, solution,...

Fig. 5. The keyword queries we selected for the Semantic Helper evaluation.

from this set of documents, we automatically generated a Semantic Glossary and obtained as a result 903 different terms extracted from the documents. Then, we considered and evaluated 100 typical queries which are usually submitted by FACIT-SME partners with reference to this collection; each query is either composed by a short text containing candidate keywords (80 queries), as in the "From Scratch" scenario of the project, or by a whole document (20 queries), according to the "From Methodology" scenario. In the following, for clarity of presentation and without loss of generality, we present the results we obtained on a reduced sample set of queries which is representative of the results we obtained on the whole set. In particular, $(Q1 - Q9)$ are the queries executed for the "From Scratch" scenario, while queries $QT1 - QT4$ are the existing enterprise documents selected for the "From Methodology" scenario. Figure 5 shows the main keywords contained in $Q1 - Q9$ queries; on the other hand, documents $QT1 - QT4$ are very large and it is not possible (nor useful to the this analysis) to summarize them to a few keywords. In the majority of the original considered queries, they were mostly constituted by specific domain terms (queries $Q1 - Q6$ capture this fact), while queries $Q7 - Q9$ are representative of less common (and ambiguous) requests. Each query has been processed by the proposed techniques so to generate a set of possible "suggestions", i.e., pointers to the relevant documents in the collection: "all-senses" will be specifically analyzed in Section 6.1, "sense-aware", in Section 6.2).

For evaluating the effectiveness of our approach, we compared the output of the Semantic Helper, for each query, with a "gold standard", i.e. the relevant answers manually selected from the set of documents by experts in the field. Two baselines representing typical syntactic retrieval methods are also considered in order to fully understand the benefits of the semantic features. Finally, Section 6.3 concludes the evaluation and is specifically devoted to analyse in detail the effectiveness of the proposed ranking and scoring techniques; therefore, it will focus on the "larger" $QT1 - QT4$ queries.

6.1. Effectiveness of All-Senses Techniques

The first analysis we conducted was to assess the quality of the all-sense semantic similarity techniques results (sense-aware extensions will be discussed in the next section) in terms of precision and recall, which are typical evaluation metrics in

Query	Our results (All-senses)			Typical Enterprise CMS Baselines					
				No sem syn/rel			No kw sel		
	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
Q1	1.000	1.000	1.000	1.000	1.000	1.000	0.011	0.420	0.022
Q2	1.000	1.000	1.000	1.000	1.000	1.000	0.005	0.330	0.010
Q3	1.000	1.000	1.000	1.000	0.969	0.984	0.946	0.240	0.383
Q4	0.947	1.000	0.973	1.000	0.079	0.146	0.921	0.321	0.476
Q5	0.878	1.000	0.935	1.000	0.077	0.143	0.986	0.235	0.380
Q6	0.923	0.949	0.936	1.000	0.333	0.500	0.967	0.369	0.534
Q7	0.839	0.837	0.838	1.000	0.642	0.782	0.901	0.421	0.574
Q8	0.313	0.781	0.447	0.712	0.303	0.425	0.653	0.288	0.400
Q9	0.203	0.688	0.313	0.652	0.043	0.081	0.467	0.029	0.055

Fig. 6. Effectiveness analysis: precision, recall and F-measure (standard results for all-senses techniques on the left, two baselines on the right).

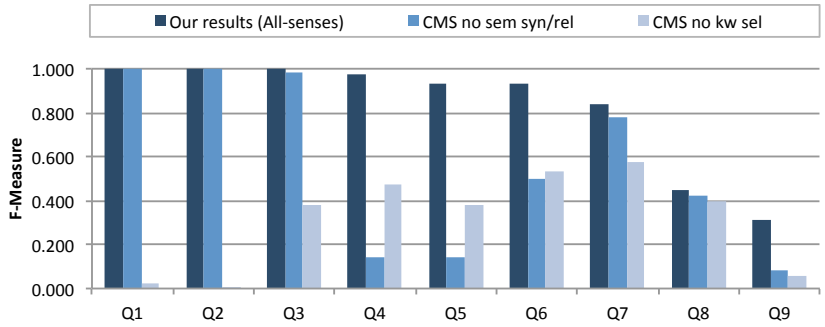

Fig. 7. Effectiveness analysis: graphical comparison for F-measure (all-senses techniques vs two baselines).

the information retrieval field[11] (Baeza-Yates and Ribeiro-Neto, 1999). Figure 6 shows the results for $Q1-Q9$ (left part of figure). The shown results are obtained with the gloss similarity as similarity function for IEEE terms and the hypernym-based one for WordNet terms. Besides precision and recall, we also report their weighted harmonic mean (F-measure).

Further, in order to emphasize the contribution and benefits of these techniques w.r.t. the ones typically available in commercial tools used by SMEs, we also present the results concerning two baselines (right part of Figure 6): (1) the syntactic retrieval method offered by most enterprise CMSs such as Alfresco, where synonyms and related terms are not handled and only exact match among terms is allowed (see also Section 5); (2) another syntactic method not exploiting the keyword enhancement phase. The comparison between the achieved F-measures is also graphically shown in Figure 7. Let us now analyze the results in detail.

In Figure 6, the precision and recall levels achieved by the semantic similarity techniques on all the queries are shown. The levels on the queries (Q1Q6) are very satisfying (equal or higher to 0.84 and 0.90, respectively). In general, the pro-

[11] Precision is defined as the fraction of retrieved documents which are known to be relevant, recall is the fraction of known relevant objects which were actually retrieved.

cessing of all queries greatly benefits from the keyword extraction/enhancement phase: in fact, without it, recall levels significantly drop to 0.2-0.4 (for instance, different inflections of the same term are not correctly identified). Keyword enhancement can also bring great benefits in precision as, for instance, in $Q1$ and $Q2$: since they contain, among others, composite expressions, such as "interface requirement" ($Q1$) or "configuration management system" ($Q2$), not correctly identifying them leads to a large number of irrelevant retrieved documents (in the second baseline, precision drops to less than 0.01, compared to 1 for the standard results). Queries $Q3-Q6$ require synonyms and related terms management in order to provide satisfying answers: for instance, one of the key terms in $Q3$ is "supplier", a concept which is expressed as "vendor" in some of the documents (recall goes from 1 to 0.96 of the first baseline), while $Q4$ contains "purpose" which is mostly expressed as "objective" in the collection (recall drops from 1 to less than 0.08). The same holds for the related terms: by applying the gloss similarity formulas exploiting the IEEE definitions, we achieve near-perfect recall levels (as opposed to the less than optimal ones of the first baseline), also maintaining high precision levels for (queries $Q1-Q6$). For example, most documents containing "review" are also relevant to $Q5$, which contains "audit"; the ones containing "document" are also relevant to $Q6$ asking for "documentation", and so on.

Queries $Q7-Q9$ obtained less satisfying levels of precision and recall, due to the presence of several common terms that are not present in the specialized IEEE vocabulary and for which only the use of WordNet similarity is allowed. Even if in some cases, as in $Q7$, the WordNet based similarity proves equally useful as the gloss based one, in other cases (as in $Q8$ and $Q9$) it leads to several false-positives. This is mainly due to the non-specialized nature of the WordNet thesaurus, which covers several domains and thus, unlike IEEE, includes highly polysemic terms (e.g., "area" and "subject"). In the next section, we will see how the sense-aware techniques can significantly improve the obtained results.

6.2. Impact on Effectiveness of Sense-Aware Techniques

In this section, we are interested in evaluating the effect of the WSD process on the semantic similarity techniques. To this end, we applied the sense-aware similarity techniques to the queries $Q1-Q9$. Figure 8 shows the results of the sense-aware similarity techniques (queries $Q6-Q9$) by comparing precision (Figure 8-a), recall (Figure 8-b) and F-Measure (Figure 8-c) to the ones obtained with the all-senses similarity techniques.

For the queries $Q1-Q6$, the sense-aware extensions behave in a similar way, with reduced improvements over the results presented in the previous section (differences in terms of 0.04 in precision or less). Thus, for sake of simplicity, we report in Figure 8 only the results for query $Q6$. The motivations are the following. Queries $Q1-Q6$ mainly contain specific and technical terms (e. g., "ISO", i.e., the acronym for "International Organization for Standardization" or "traceability") existing only in the IEEE vocabulary and not in WordNet. As a consequence, these terms are typically not affected by disambiguation issues (see Section 4). Moreover, these queries contain several terms existing in WordNet but with a unique meaning (i.e., monosemic terms, e.g., "software" or "identifier") or with few strongly related meanings (e.g., "methodology", "hardware", or "recovery").

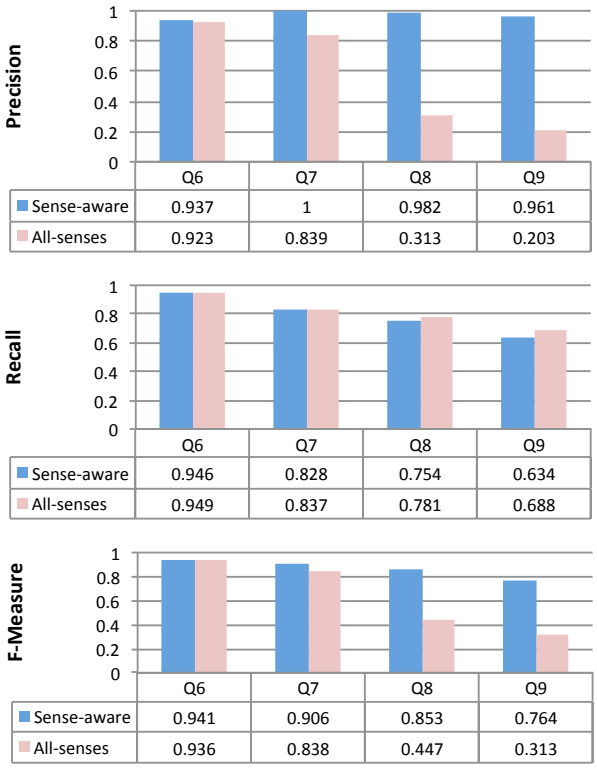

Fig. 8. Improvements achieved by the sense-aware techniques over the all-senses ones: a) precision, b) recall, c) F-measure.

On the contrary, the use of WSD is able to significantly improve the accuracy of the document retrieval process for queries $Q7 - Q9$ (see Figure 8), for which we obtained an average increment in precision of 0.53. The main reason for these results is that these queries are more ambiguous than $Q1 - Q6$ as they contain general terms, which might be related to several documents. For instance, query $Q9$ greatly benefits from WSD, as it contains, among others, the term "area": this term may mean a "geographical region" (as in $Q9$) or "a subject of study". ORM documents are usually not about "area" as "geographical region", while in several ones "area" has the "a subject of study" sense. Without disambiguation, we obtained several false positive documents corresponding to "area" with the "subject of study" sense and other documents containing false related terms, such as "topic", "issue" and "subject". The same happens for query $Q8$ containing, among others, the general and polisemic term "scheme", which can assume several meanings, such as a "strategy", a "dodge", a "system" etc.

As regards to query $Q7$, by using WSD we were able to achieve perfect precision. In this case, the improvement w.r.t. the all-sense technique is of 0.16, which is a smaller increment than for $Q8$ and $Q9$: this is due to the fact that, even if the query contains common terms, their meaning is the same as in the majority of their occurrences in the ORM documents. For instance, in $Q7$, the

Query	Synonyms Sense-aware			Synonyms All-senses		
	Prec	Rec	F	Prec	Rec	F
Q7	1.000	0.886	0.940	1.000	0.886	0.940
Q8	1.000	0.579	0.733	0.552	0.619	0.584
Q9	1.000	0.063	0.119	0.293	0.063	0.104

Query	Semantically rel. terms Sense-aware			Semantically rel. terms All-senses		
	Prec	Rec	F	Prec	Rec	F
Q7	1.000	0.837	0.911	0.839	0.837	0.838
Q8	0.982	0.769	0.863	0.312	0.778	0.445
Q9	0.961	0.568	0.714	0.203	0.575	0.300

Fig. 9. Impact of WSD on the specific aspects of semantic similarity: use of synonyms (top) and use of semantically related terms (bottom).

term "market" is used in the "commercial activity" meaning: the ORM document collection includes several documents containing "market" with this meaning.

On the other hand, as we can note from Figure 8-b, recall is only slightly affected by WSD: we obtained very similar results to the all-senses ones, with a decrease of 0.093. The reason is that WSD mainly performs a pruning action w.r.t. the document collection: in nearly all cases, the documents containing the query terms with the correct sense are preserved, while the irrelevant ones are correctly pruned out (this effect will be even clearer in the ranking analysis we provide in the next section).

We are now interested in investigating the impact of the WSD process on the single similarity techniques composing the method: in particular, we will investigate how WSD affects the use of synonyms and the use of related terms in the document similarity computation. We focus on $Q7 - Q9$. Figure 9 compares the results obtained by the sense-aware (on the left) and the all-senses (on the right) similarity techniques, by using only synonyms (top of figure) and only semantically related terms (bottom). We can note that both synonyms and related terms greatly benefit from the use of WSD. In particular, while recall is again almost unchanged (with an average decrease of 0.09), precision is strongly improved (with an average improvement of 0.46). The unique exception is represented by $Q7$ for which we obtained a slight improvement in using WSD with related terms. As previously observed, this result is due to the fact that $Q7$ represents the case of queries containing ambiguous terms that are present in the document collection with a unique meaning.

In conclusion, by analyzing the overall performances obtained by using the sense-aware similarity techniques, we can observe that, independently from the ambiguity of query terms, we can safely use WSD, as it improves precision (w.r.t. the all-senses techniques) without significantly decreasing recall. Moreover, as we will see in detail in the next section, WSD helps in positioning the most relevant documents among the top documents in the ranking.

6.3. Detailed Ranking Effectiveness Evaluation

In this section, we will deepen the effectiveness analysis by considering queries $QT1 - QT4$, in the form of actual text documents typically used in the FACIT-

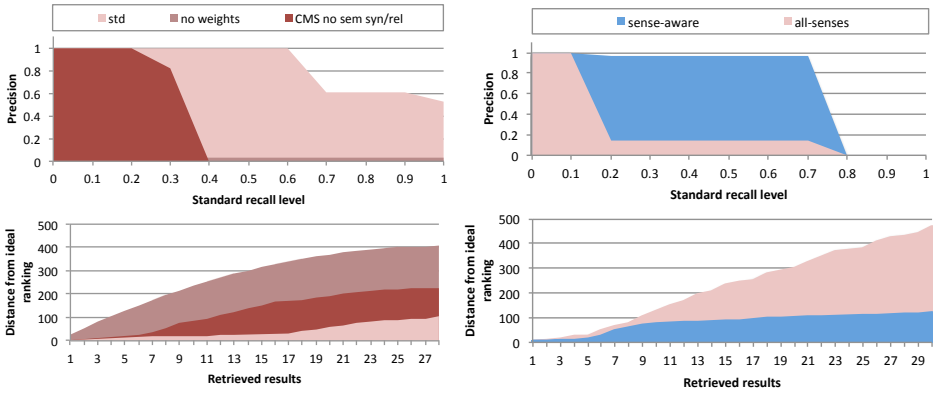

Fig. 10. In-depth analysis for all-senses techniques (QT1, on the left) and for sense-aware vs all-senses (QT4, on the right): precision at standard recall levels (top) and distance from optimal ranking (bottom)

SME environment, for which to find related documents in the collection. Differently from $Q1 - Q9$, these queries may contain a large number of terms and produce a very large number of results. For this reason, it is essential to evaluate not only which answers are returned but also their scores and the induced ranking, assessing whether the best suggestions are returned in the top positions and, thus, whether the proposed weighting scheme is effective.

We start this analysis by assessing the impact of the use of weights and synonyms/related terms on the all-senses techniques described in Section 3.3 (Figure 10). To this end, we consider queries $QT1$ and $QT2$, whose majority terms are very specific and technical (as in queries $Q1 - Q6$). In Figure 10 (left part) we show, for $QT1$ and all-senses techniques, the precision values obtained at different recall levels, i.e., when a given percentage of relevant documents have been found (top), and the distance from the ideal ranking (bottom). The all-senses technique is compared to two baselines: a non-weighted version of the $DSim$ which does not consider the term weights (i.e., essentially, they are fixed to 1) and the non-semantic CMS-like retrieval method which does not consider synonyms and related terms. Notice that the all-senses technique achieves high precision levels even at high recall levels: for instance, at recall level 0.6, the precision is still 1, while the baselines' precision levels have already dropped lower than 0.03. This confirms that our techniques are able to identify the most significant terms in the queries, without being misled by non-relevant ones. The optimal ranking distance analysis confirms the goodness of the retrieved results: for each alternative, the curve represents the normalized Spearman footrule distance (Diaconis and Graham, 1977) between the retrieved and the ideal ranking, i.e., the normalized sum of the absolute values of the difference between the ranks. For QT2 we found an equally good performance with very similar graphs, therefore we will not show them in detail.

Finally, we were interested in evaluating how the sense-aware techniques contribute to a highly effective ranking. We considered documents $QT3$ and $QT4$, containing a large percentage of common/ambiguous terms, which are well suited to stress the effectiveness of the proposed techniques on difficult requests. The

right part of Figure 10 shows the results we obtained for $QT4$ ($QT3$ showed similar trends): as we can see, the precision is kept high even at high recall levels, and the distance from optimal ranking is kept much lower than for the all-senses techniques. This once again shows the positive impact of the WSD technique, which gives a key contribution in retrieving the most relevant results among the first ones in the ranking.

7. Concluding Remarks

In this paper we proposed a fully automatic and semantic approach for filtering/searching software documentation, carefully considering the actual user-targets, IT-SME.

Our approach has been studied and evaluated in the context of the European FP7 3-years project "Facilitate IT-providing SMEs by Operation-related Models and Methods (FACIT-SME)", and is implemented in the Semantic Helper component of the FACIT-SME solution.

The main achievements and effectiveness of the proposed approach, as documented by the experimental section, are the following:

- Firstly, it is an easyly and generally applicable/maintainable tool for IT SMEs to look for information in their large amounts of already available textual documents (i.e., methodology descriptions, and so on); this is done by specifying simple keyword queries or document queries, without requiring any conversion towards complex structured formats which would be time and cost consuming;
- Secondly, the tool is able to automatically identify the similarities between such queries and a reference set of documents. The limitations of standard syntactical techniques (such as the ones usually exploited by enterprise CMSs) are overcome by considering the semantics intrinsically associated to document/query terms and by addressing the problem of term ambiguity through the use of WSD algorithms. To this aim, we exploited different kinds of external knowledge sources (both general and specic domain dictionaries or thesauri);
- Thirdly, its flexibility enables many high-level functionalities, i.e., filtering software methodologies for software process assessment and improvement, quality requirements for helping in certication process, best practices for facilitating knowledge sharing, and so on;
- Finally, the approach does not have any prerequisite, such as the knowledge of complex formal representation/querying standards or the need of converting/updating the documentation already available in the enterprise. To this end, our proposal leverages on the strengths of both classic information retrieval and of knowledge-based techniques, without impairing general applicability and usability.

Several paths will be contemplated as future work:

- We will further analyze and refine the similarity techniques, user feedback on the retrieved suggestions, multilanguage information management and querying support;
- We will investigate how techniques we developed in complementary contexts, such as multi-version semi-structured data management (Grandi, Mandreoli, Martoglia, Ronchetti, Scalas and Tiberio, 2008), could help in the exploitation of other non-textual knowledge available in the ORM repository;

– Leveraging on our previous works on Peer-to-Peer network (Beneventano, Bergamaschi, Guerra and Vincini, 2005; Mandreoli, Martoglia, Penzo and Sassatelli, 2009), we will extend our approach in this direction. Indeed, large software development projects are complex endeavors that involve numerous participants which can work across several sites and act in various roles; therefore, we will also consider notable past experiences such as (Happel, Maalej and Stojanovic, 2008), where a Peer-to-Peer based metadata management semantic technologies was proposed as an important enabler to improve information and knowledge sharing in such scenarios.

8. Acknowledgment

The research leading to these results has received funding from 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]) under grant agreement n. 243695.

Our sincere thanks to Domenico Beneventano (UniMoRe), Gorka Benguria (ESI), Frank-Walter Jaekel (Fraunhofer IPK) and to the other project partners for their support to this research.

References

Aetic (Spain) and Agoria (Belgium) and AssInform (Italy) et. al. (24 October 2008), *in* 'Position paper towards a European software strategy presented to commissioner Viviane Reding'.

Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999), *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Banerjee, S. and Pedersen, T. (2003), Extended Gloss Overlaps as a Measure of Semantic Relatedness, *in* G. Gottlob and T. Walsh, eds, 'IJCAI', Morgan Kaufmann, pp. 805–810.

Beneventano, D., Bergamaschi, S., Guerra, F. and Vincini, M. (2001), The momis approach to information integration., *in* '3rd International Conference on Enterprise Information Systems (ICEIS)', pp. 194–198.

Beneventano, D., Bergamaschi, S., Guerra, F. and Vincini, M. (2005), Querying a super-peer in a schema-based super-peer network., *in* 'DBISP2P', Vol. 4125 of *Lecture Notes in Computer Science*, Springer, pp. 13–25.

Bergamaschi, S., Martoglia, R. and Sorrentino, S. (2012), A semantic method for searching knowledge in a software development context., *in* 'SEBD', pp. 115–122.

Binkley, D. and Lawrie, D. (2010), Maintenance and evolution: Information retrieval applications, *in* P. A. Laplante, ed., 'Encyclopedia of Software Engineering', Taylor & Francis, pp. 454–463.

Carnegie Mellon University Software Engineering Institute (2006), *in* 'CMMI for Development, Version 1.2 (pdf)'.

Constantopoulos, P., Jarke, M., Mylopoulos, J. and Vassiliou, Y. (1995), 'The software information base: A server for reuse', *VLDB Journal* **4**, 1–43.

DG INFSO Internal Reflection Group on Software Technologies, ITEA (April 2002).

Diaconis, P. and Graham, R. L. (1977), 'Spearman's footrule as a measure of disarray', *Royal Statistical Society Series B* **32**(24), 262–268.

DIS 9001:2000 Quality Management Systems - Requirement (pdf) (1999), *in* 'ISO TC176'.

FACIT-SME Project (2010-2012), 'Facilitate IT-providing SMEs by Operation-related Models and Methods'.
URL: *http://www.facit-sme.eu/*

Gall, C. S., Lukins, S. K., Etzkorn, L. H., Gholston, S., Farrington, P., Utley, D. R., Fortune, J. and Virani, S. (2008), 'Semantic software metrics computed from natural language design specifications.', *IET Software* **2**(1), 17–26.
URL: *http://dblp.uni-trier.de/db/journals/iee/iet-s2.htmlGallLEGFUFV08*

Garg, A., Goyal, D. P. and Lather, A. S. (2010), 'The influence of the best practices of information system development on software smes: a research scope.', *IJBIS* **5**(3), 268–290.
URL: *http://dblp.uni-trier.de/db/journals/ijbis/ijbis5.htmlGargGL10*

Girardi, M. R. and Ibrahim, B. (1994), A similarity measure for retrieving software artifacts, *in* 'SEKE', Knowledge Systems Institute, pp. 478–485.

Grandi, F., Mandreoli, F., Martoglia, R., Ronchetti, E., Scalas, M. R. and Tiberio, P. (2008), Ontology-based personalization of e-government services, *in* 'Intelligent User Interfaces: Adaptation and Personalization Systems and Technologies, Constantinos Mourlas and Panagiotis Germanakos (Ed.), IGI Global', pp. 167–187.

Gyimothy, T., Ferenc, R. and Siket, I. (2005), 'Empirical validation of object-oriented metrics on open source software for fault prediction', *IEEE Trans. Softw. Eng.* **31**(10), 897–910.
URL: *http://dx.doi.org/10.1109/TSE.2005.112*

Happel, H.-J., Korthaus, A., Seedorf, S. and Tomczyk, P. (2006), Kontor: An ontology-enabled approach to software reuse, *in* K. Zhang, G. Spanoudakis and G. Visaggio, eds, 'SEKE', pp. 349–354.

Happel, H.-j. and Seedorf, S. (2006), 'Applications of ontologies in software engineering', *Engineering* pp. 1–14.
URL: *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.5733rep=rep1type=pdf*

Happel, H., Maalej, W. and Stojanovic, L. (2008), Team: towards a software engineering semantic web, *in* 'Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2008, Leipzig, Germany, Tuesday, May 13, 2008', pp. 57–60.

InnoSME Project (2008), InnoSME Project, a support action of the ICT program.
URL: *http://cordis.europa.eu/news/rcn/28963$_e$n.html*

Kiefer, C., Bernstein, A. and Tappolet, J. (2007), Mining Software Repositories with iSPAROL and a Software Evolution Ontology, *in* 'MSR', IEEE Computer Society, p. 10.

Leacock, C. and Chodorow, M. (1998), *Combining Local Context and WordNet Similarity for Word Sense Identification*, The MIT Press, chapter 11, pp. 265–283.

Lethbridge, T. C., Singer, J. and Forward, A. (2003), 'How software engineers use documentation: The state of the practice', *IEEE Softw.* **20**(6), 35–39.
URL: *http://dx.doi.org/10.1109/MS.2003.1241364*

Leung, H., Liao, L. and Qu, Y. (2005), A software process ontology and its application, *in* 'the 4th International Semantic Web Conference'.

Mandreoli, F. and Martoglia, R. (2011), 'Knowledge-based sense disambiguation (almost) for all structures', *Inf. Syst.* **36**(2), 406–430.

Mandreoli, F., Martoglia, R., Penzo, W. and Sassatelli, S. (2009), 'Data-sharing p2p networks with semantic approximation capabilities', *IEEE Internet Computing (IEEE)* **13**(5), 60–70.

Mandreoli, F., Martoglia, R. and Ronchetti, E. (2005), Versatile structural disambiguation for semantic-aware applications, *in* O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury and W. Teiken, eds, 'CIKM', ACM, pp. 209–216.

Martoglia, R. (2011), Facilitate IT-Providing SMEs in Software Development: a Semantic Helper for Filtering and Searching Knowledge, *in* 'SEKE', pp. 130–136.

Miller, G. A. (1994), Wordnet: A lexical database for english, *in* 'HLT', Morgan Kaufmann.

Mylopoulos, J., Borgida, A., Jarke, M. and Koubarakis, M. (1990), 'Telos: representing knowledge about information systems', *ACM Trans. Inf. Syst.* **8**(4), 325–362.
URL: *http://doi.acm.org/10.1145/102675.102676*

Navigli, R. (2009), 'Word sense disambiguation: A survey', *ACM Comput. Surv.* **41**(2).

ORM Architecture and Engineering Models (October 2010), *in* F. W. Jaekel, ed., 'FP7-SME FACIT-SME (FP7-243695), Deliverable'.

OSES Architecture and Component Specification (December 2010), *in* G. Benguria, ed., 'FP7-SME FACIT-SME (FP7-243695), Deliverable'.

Palmer, M., Dang, H. T. and Fellbaum, C. (2007), 'Making fine-grained and coarse-grained sense distinctions, both manually and automatically', *Natural Language Engineering* **13**(2), 137–163.

Po, L. and Sorrentino, S. (2011), 'Automatic generation of probabilistic relationships for improving schema matching', *Inf. Syst.* **36**(2), 192–208.

Poshyvanyk, D. and Marcus, A. (2006), The conceptual coupling metrics for object-oriented systems, *in* 'Proceedings of the 22nd IEEE International Conference on Software Maintenance', ICSM '06, IEEE Computer Society, Washington, DC, USA, pp. 469–478.
URL: *http://dx.doi.org/10.1109/ICSM.2006.67*

Salton, G. and Buckley, C. (1988), 'Term-Weighting Approaches in Automatic Text Retrieval', *Inf. Process. Manage.* **24**(5), 513–523.

Shepherd, D., Fry, Z. P., Hill, E., Pollock, L. L. and Vijay-Shanker, K. (2007), Using natural language program analysis to locate and understand action-oriented concerns, *in* B. M. Barry and O. de Moor, eds, 'AOSD', Vol. 208 of *ACM International Conference Proceeding Series*, ACM, pp. 212–224.

Sorrentino, S., Bergamaschi, S. and Gawinecki, M. (2011), NORMS: An automatic tool to perform schema label normalization, *in* S. Abiteboul, K. Böhm, C. Koch and K.-L. Tan, eds, 'ICDE', IEEE Computer Society, pp. 1344–1347.

Soydan, K. (2006), 'An owl ontology for representing the cmmi-sw model', *2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006) at ISWC 06* . **URL:** *http://km.aifb.uni-karlsruhe.de/ws/swese2006/final/soydan-full.pdf*

Sridhara, G., Hill, E., Pollock, L. and Vijay-Shanker, K. (2008), Identifying word relations in software: A comparative study of semantic similarity tools, *in* 'Proceedings of the 2008 The 16th IEEE International Conference on Program Comprehension', ICPC '08, IEEE Computer Society, Washington, DC, USA, pp. 123–132. **URL:** *http://dx.doi.org/10.1109/ICPC.2008.18*

Standish Group (2006), *in* 'Chaos Report 2006'.

Trakarnviroj, A. and Prompoon, N. (2012), A storage and retrieval of requirement model and analysis model for software product line, *in* 'Proceedings of the International MultiConference of Engineers'.

Udomchaiporn, A., Prompoon, N. and Kanongchaiyos, P. (2006), Software requirements retrieval using use case terms and structure similarity computation, *in* 'Proceedings of the XIII Asia Pacific Software Engineering Conference', APSEC '06, IEEE Computer Society, Washington, DC, USA, pp. 113–120. **URL:** *http://dx.doi.org/10.1109/APSEC.2006.53*

Varghese, M. and Systems, C. (2012), 'Content strategy for small and medium enterprises (smes)', *Best Practices* **14**(5).

Witte, R., Zhang, Y. and Rilling, J. (2007), Empowering software maintainers with semantic web technologies, *in* 'Proceedings of the 4th European Conference on The Semantic Web: Research and Applications', ESWC '07, Springer-Verlag, Berlin, Heidelberg, pp. 37–52. **URL:** *http://dx.doi.org/10.1007/978-3-540-72667-8*$_5$