Knowledge Discovery from Textual Sources by using Semantic Similarity*

Paolo Atzeni[1], Fabio Polticelli[2], and Daniele Toti[1]

[1] Department of Computer Science and Automation, Roma Tre University
atzeni@dia.uniroma3.it, toti@dia.uniroma3.it
[2] Department of Biology, Roma Tre University
polticel@uniroma3.it

Abstract. We propose a methodology to automatically discover characterizing knowledge from textual sources, with the purpose of semantically categorizing them and clustering them together according to their subjects. Such a methodology is based upon several challenging steps, like terminology extraction and disambiguation, semantic similarity identification via ontology alignment, and a core pattern-based strategy for automatic ontology building. This methodology was originally devised as an extension of PRAISED, our abbreviation identification and resolution proposal, with the purpose of allowing us to resolve previously unresolvable abbreviations, whose explanation either escapes the system's proximity-based approach or is not found within the very source text they are featured in. By moving from a paper-by-paper, mainly syntactical process to a corpus-based, semantic approach, it will be in fact possible to dramatically enhance our system in terms of its resolution capabilities. Nevertheless, the strategy we present here is not tied to this specific task, but is instead of relevance for a variety of contexts, and might therefore find a far wider applicability for other advanced knowledge extraction and discovery systems.

1 Introduction

Textual documents, either in their purely unstructured or semi-structured form, are undoubtedly the repository of most of the human knowledge. As the amount of available digital information grows to previously unimaginable levels, an unprecedented number of documents containing essential knowledge, albeit scattered among them, is at people's disposal for a variety of different studies and researches. All of the everyday manual operations involved for collecting and organizing such a knowledge are consequently getting more and more painstaking and time-consuming, thus yearning for semi-automatic or automatic solutions to help reduce costs in terms of both time and employed resources.

In this paper, we propose a methodology to automatically discover characterizing knowledge from scientific papers (full-texts), in order to sort them out according to the subjects they discuss, and therefore allow for speed-reading and

* Extended Abstract

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

cataloging activities. This methodology is born as an extension of PRAISED [2–4, 13], our information extraction system whose main purpose is identifying and resolving abbreviations from full-text scientific papers. This abbreviation discovery process, as originally implemented, proceeded on a paper-by-paper basis, by deeming an abbreviation resolvable if and only if its explanation could be found within the same paper the former was featured in. Besides, a proximity-based scan was used to check for abbreviation explanations, thus potentially failing in resolving abbreviations whose explanation is mentioned in a whole difference section of the considered paper. By computing semantic similarity among papers from a given domain, we will show how it is possible to shift from a paper-by-paper to a corpus-based approach, so that abbreviations found in a determined text might be resolved by drawing on a different paper sharing similar subjects with the first. The strategy we discuss is made up of several substeps, involving tasks like terminology extraction, terminology disambiguation according to context, computation of semantic distance and automatic ontology building/learning and matching/alignment. It must be stressed out that such a strategy, despite being applied for the purpose of expanding the abbreviation discovery capabilities of our system, is relevant for a wider range of applications, and might be used as successfully to enhance other knowledge extraction systems as well.

The paper is structured as follows. In Section 2, related work is discussed. In Section 3, we briefly describe our abbreviation discovery system. In Section 4, we delve into the details of our strategy, by discussing its required steps and their expected outcome. And finally, in Section 5, we draw our conclusions.

2 Related Work

As far as abbreviation discovery is concerned, several research groups have proposed a certain number of methodologies, ranging from general approaches to more specific techniques. These include the use of regular expressions, linguistic cues and pattern-based recognition strategies, as well as machine learning algorithms, natural language processing and mixed methods. Even most of the major methods among them [10], [7], [1] possess several limitations (strong constraints, abstract-scope only, limited recall, no entity recognition etc.).

Regarding research areas like ontology building/learning and ontology matching/alignment, literature is vast and several potential approaches have been devised. A survey on ontology building methodologies is shown in [12]. Relevant proposals for ontology learning can be found in [14] and [5], although either they require human intervention or they show practical limitations even when dealing with simpler and shorter texts.

As far as ontology matching is concerned, [6] presents a thorough classification of ontology alignment methodologies and tools. One such tool is COMA [8], based on a decade-old experience in the field and largely used by the scientific community.

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

3 The PRAISED System

PRAISED (currently defined as Processor for Abbreviation Identification, Disambiguation and Storage) was initially designed and implemented as an abbreviation discovery system for the biomedical community, focusing on the identification and resolution of protein abbreviations from full-text biological papers. Recently, it has been refined and generalized, by successfully applying it to other, non-biomedical domains. Further details can be found in [2–4, 13].

4 Knowledge Discovery Strategy

In this section we will provide the details for our knowledge discovery methodology, as composed by the three main steps further discussed below.

4.1 Step 1: Finding characterizing elements in a paper

The first step of the strategy takes as input a corpus of full-text papers, all sensibly belonging to a certain known domain, and produces as output an ontology for each of the papers making up the corpus: the ontology will represent the characterizing concepts detected from the text. This step, which involves several computationally-heavy tasks, is performed only once and one paper at a time, as a prerequisite for the subsequent steps, which are instead executed on demand.

Terminology extraction, disambiguation and classification In order to discover the characterizing elements of the considered paper, relevant terms must be extracted from the source text. This is a natural language processing task, where techniques like part-of-speech (POS) tagging, stemming and lemmatization play a central role. A sample text excerpt can be found in the leftmost part of Figure 1, taken from one of the Wikipedia entries for "Java", and will be used as an example throughout this discussion. We will refer to it as Paper 1.

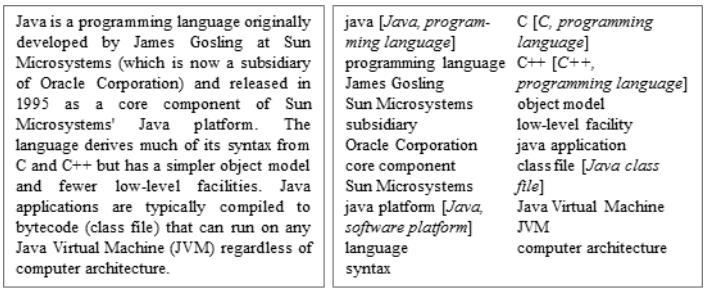

Fig. 1. A text excerpt (on the left) and its characterizing terms (on the right)

The operation pipeline is as follows:

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

- **POS tagging**. The source text is tagged in order to identify the POS elements within it. For our examples, we use the Stanford NLP tools [11].
- **Candidate characterizing term selection and aggregation**. After the tagging is done, nouns, proper nouns and their combination with adjectives are selected as preliminary candidate characterizing terms. Adjacent nouns are combined as well to represent a single compound term.
- **Lemmatization**. Candidate terms from the previous selection are brought back to their lemma form, in order to ease the disambiguation to follow.
- **Terminology disambiguation and filtering**. Candidates thus identified are disambiguated according to words adjacent or next to them, so that polysemous terms are correctly associated with their actual meaning with respect of the lexical context they are placed in. This activity requires an external knowledge base: we take advantage of Wikipedia Miner for this purpose. The resulting candidate term list of Paper 1 after selection, aggregation, lemmatization and disambiguation is shown in the rightmost part of Figure 1.
- **Terminology classification**. Once the context information has been correctly associated with the candidate terms, they are given a score, or weight, based on the term frequency (TF); only those scoring higher than a set threshold will be actually selected. In the case of our sample text, due to its short length the majority of the detected terms share more or less the same score, for they appear only once in the text (with a few exceptions); the threshold setting is thus responsible to take into account the text's length, so that the classification procedure can work well with differently-sized input papers.
- **Abbreviation identification**. Along with the term list produced so far, a list of the abbreviations featured within the considered paper is also compiled and stored, by taking advantage of Phase 1 of PRAISED's discovery process. In Paper 1, only the term *JVM* is recognized as an abbreviation, and thus stored separately from the characterizing terms earlier produced (even though also featured among them).

Ontology building The next challenging task lies in tying together the characterizing terms extracted so far, by identifying relationships among them. This is what we mean with ontology learning or building, and comes down to tracing back, in an automatic fashion, explicit (or somewhat implicit) ties among terms.

In this regard, a pattern-matching strategy is employed to discover a certain number of interrelationships from the lexical contexts of the considered terms. Obviously, this kind of automatic detection could not claim completeness: only a selected number of relationships might be inferred in this fashion (e.g. is-a, equivalence, part-whole, property/relation etc.). On the other hand, by restricting the range of relationships to a similar subset, the semantic comparison to be performed in Step 2 can be smoothed and produce more effective results.

Let us review this process by considering our example. In Figure 2 we see the resulting ontology automatically built from the source text categorized in Step

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

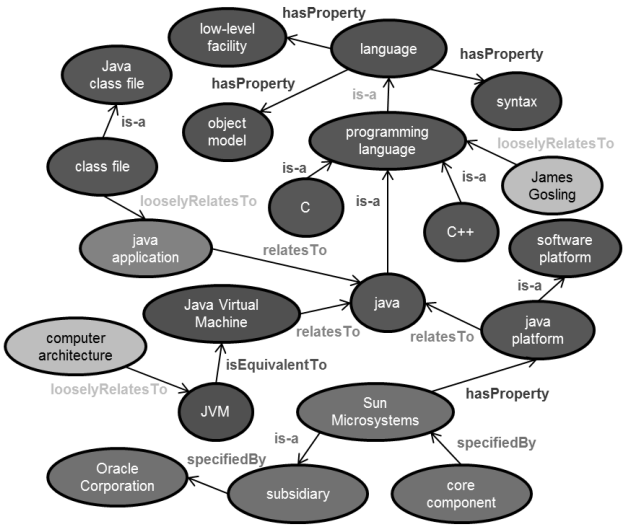

Fig. 2. Ontology automatically built from the characterizing terms obtained in Step 1 for Paper 1

1. Relationships correlating the characterizing terms of the input text are built in the following order and with the following strategies (each step corresponding to a different relationship color in the figure; the background color of an element underlines the substep in which that element first appears in the ontology):

- is-a relationships derived from terminology disambiguation (blue color);
- equivalence relationships, via equivalence patterns (following parenthesis-enclosed explanation, correlation expressions like "as known as" etc.) (purple color);
- is-a relationships from lexical patterns (verb "to be" + article, relative connectors + verb "to be", "such as" etc.), and specification relationships (from "of" connectors) (green color);
- part-of relationships from expressions like "is made of" etc., and property-owning relationships from verb "to have", possessive adjectives and similar expressions (brown color);
- is-a relationships derived from lexical inclusion of characterizing terms (as in "programming language", which is a "language") (olive color);
- general relationships from terms sharing an adjective/a specifying element to the actual term shared (as in "Java Virtual Machine", "java platform", "java application", which are all related to "java") (light blue color);
- loose relationships for tying terms either isolated or yet to be correlated, according to proximity and appearance in the same sentence (the uncorrelated "computer architecture" to "class file", or the isolated "class file" to "java application") (yellow color).

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

Some imprecisions can be noticed in the automatic building. For instance, "class file", earlier disambiguated as "java class file", ends up being a subclass of "java class file", whereas it should be the other way around; also, a significant term like James Gosling, the Java creator, gets loosely tied to "programming language" instead of "java", due to the inability of inferring a specific relationship between it and the term "java". There are of course margins of improvement for the automatic ontology building process based on lexical patterns.

In the end, though, the results of Step 1 will be as many ontologies as the papers processed, along with an abbreviation list for each of them. This way, a full-text corpus can be automatically categorized with semantic information for the papers it includes.

4.2 Step 2: Computing semantic similarity

Once the paper corpus has been properly semantically categorized, it is possible to proceed with the on-demand steps. The second step takes place whenever a paper, scanned by PRAISED for abbreviations, ends up featuring an abbreviation without a corresponding explanation. This may happen for two reasons: either the abbreviation explanation escapes the proximity-based approach implemented by PRAISED (ending up in a different sentence or a different section of the document altogether), or it is simply not present within that very paper. In order to try and resolve such an unresolved abbreviation nevertheless, we need to identify those papers bearing the highest similarity with the original text in terms of the subjects discussed. For this purpose, the ontologies created in Step 1 must be purposefully compared.

A Java Virtual Machine is a piece of software that is implemented on non-virtual hardware and on standard operating systems. A JVM provides an environment in which Java bytecode can be executed, enabling such features as automated exception handling, which provides "root-cause" debugging information for every software error (exception).	Java Virtual Machine piece software [*Computer software*] non-virtual hardware standard operating system JVM environment java bytecode feature automated exception handling root-cause debugging information software error [*Software bug*] exception

Fig. 3. A text excerpt featuring an unresolvable abbreviation (on the left) and its characterizing terms (on the right)

Let us consider the text in Figure 3, taken from the Wikipedia entry for "Java Virtual Machine", which we will refer to as Paper 2, along with its characterizing terms. Such an excerpt features an abbreviation, *JVM*, which escapes PRAISED's proximity approach for its abbreviation resolution phase: thus, it cannot be resolved by the system. This text turns into the ontology in Figure 4 after Step 1. Incidentally, such an ontology features an island of terms isolated from the rest of the structure: this might not be allowed in certain formalizations.

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

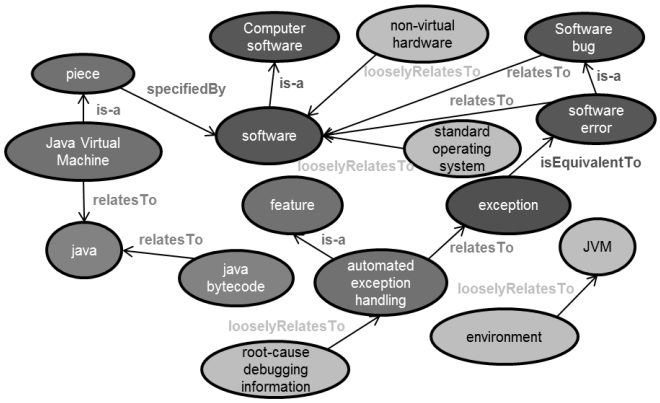

Fig. 4. Resulting ontology for Paper 2

Ontology alignment The process of comparing ontologies with each other is usually defined as ontology matching or alignment. Given two ontologies, it is paramount to try and identify terms which are both alike and related to all or some of the same concepts as well. For this purpose, we first rely onto the lexical level, by checking the mutual distance among the terms of the source and target ontology, via the Jaccard and weighted edit distance; afterwards, we focus on the ontological structure, and compute a comparison in terms of the kinds of relationships defined between elements from the considered ontologies. The result of this matching returns a similarity score: those papers deemed most similar will be the candidates where unresolvable abbreviations from the original paper might be indeed found. During this phase, the abbreviation lists from Step 1 is also taken into consideration: the amount of abbreviations shared between papers affects the final similarity score, which is increased accordingly. In the example proposed, Paper 2 is compared to the other papers in the corpus in terms of their ontologies. The ontology built from Paper 2 and the one built from Paper 1 are eventually assigned a high similarity score, since they share several similarities at various levels.

4.3 Step 3: Tracing back abbreviation explanations

The final step is simply a matter of applying the PRAISED process to the papers most similar to the one considered. From our example, Paper 1, once assigned a high similarity score with respect of Paper 2, will be a fit candidate to be used as the search space for the unresolved abbreviation (*JVM*) found in Paper 2. Indeed, the explanation of such an abbreviation is actually featured in Paper 1 (and appears in its abbreviation list as well), and will be successfully matched with its corresponding abbreviation by PRAISED's resolution process, this way resolving the original unresolved abbreviation from Paper 2.

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

5 Conclusions

In this paper we have proposed a methodology to discover characterizing knowledge from a corpus of full-text papers, in order to assess their semantic proximity and be able to categorize them according to similar topics. This strategy widens the scope of our abbreviation discovery system to a corpus-based approach, where an abbreviation explanation may be detected in any other, semantically similar text from a given corpus. At the same time, this methodology might in principle be applied to different contexts as well, for no constraints tie it to the specific task at all. At the present time, we are finalizing its implementation: some candidate tools have already been sorted out, integrated and tested for several of its substeps, whereas we are currently fine-tuning and experimenting our core ontology building algorithm. Minor refinements notwithstanding, we believe that such a methodology will end up representing a step forward both in ours and in other knowledge extraction systems.

References

1. H. Ao and T. Takagi. Alice: an algorithm to extract abbreviations from medline. In *J. Am. Med. Inform. Assoc.*, 12, pages 576-586, 2005.
2. P. Atzeni, F. Polticelli and D. Toti. A Framework for Semi-Automatic Identification, Disambiguation and Storage of Protein-related Abbreviations in Scientific Literature. In *ICDE, DaLi Workshop*, 2011.
3. P. Atzeni, F. Polticelli and D. Toti. An Automatic Identification and Resolution System for Protein-related Abbreviations in Scientific Papers. In *EvoBio*, 2011.
4. P. Atzeni, F. Polticelli and D. Toti. Experimentation of an Automatic Resolution Method for Protein Abbreviations in Full-Text Papers. In *ACM-BCB*, 2011.
5. P. Cimiano and J. Vlker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In *NLDB*, 2005.
6. J. Euzenat et al. State of the art on ontology alignment. In *Knowledgeweb*, 2004.
7. C. Kuo, M. HT Ling, K. T. Lin and C. N. Hsu. BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature. In *BMC Bioinformatics*, Vol. 10, 2009.
8. S. Massmann, S. Raunich, D. Aumueller, P. Arnold and E. Rahm. Evolution of the COMA Match System. In *OM-2011*, 2011.
9. D. Milne and I.H. Witten. An open-source toolkit for mining Wikipedia by: D. Milne, and I.H. Witten. In *NZCSRSC*, Vol. 9, 2009.
10. A. Schwartz and M. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical texts. In *PSB*, 2003.
11. The Stanford NLP Group. http://nlp.stanford.edu/index.shtml
12. R. Subhashini and J. Akilandeswari. A survey on ontology construction methodologies. In *IJECBS (Online)*, Vol. 1, No. 1, 2011.
13. D. Toti, P. Atzeni and F. Polticelli. Automatic Protein Abbreviations Discovery and Resolution from Full-Text Scientific Papers: The PRAISED Framework. In *Bio-Algorithms and Med-Systems (BAMS)*, Vol. 8, No. 1, 2012.
14. Y. Wang, J. Vlker and P. Haase. Towards Semi-automatic Ontology Building Supported by Large-scale Knowledge Acquisition. In *AAAI Fall Symposium On Semantic Web for Collaborative Knowledge Acquisition*, Vol. FS-06-06, pages 70-77, 2006.

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