Multilingual WSD with Just a Few Lines of Code: the BabelNet API

Roberto Navigli and **Simone Paolo Ponzetto**
Dipartimento di Informatica
Sapienza Università di Roma
{navigli,ponzetto}@di.uniroma1.it

Abstract

In this paper we present an API for programmatic access to BabelNet – a wide-coverage multilingual lexical knowledge base – and multilingual knowledge-rich Word Sense Disambiguation (WSD). Our aim is to provide the research community with easy-to-use tools to perform multilingual lexical semantic analysis and foster further research in this direction.

1 Introduction

In recent years research in Natural Language Processing (NLP) has been steadily moving towards multilingual processing: the availability of ever growing amounts of text in different languages, in fact, has been a major driving force behind research on multilingual approaches, from morpho-syntactic (Das and Petrov, 2011) and syntactico-semantic (Peirsman and Padó, 2010) phenomena to high-end tasks like textual entailment (Mehdad et al., 2011) and sentiment analysis (Lu et al., 2011).

These research trends would seem to indicate the time is ripe for developing methods capable of performing semantic analysis of texts written in any language: however, this objective is still far from being attained, as is demonstrated by research in a core language understanding task such as Word Sense Disambiguation (Navigli, 2009, WSD) continuing to be focused primarily on English. While the lack of resources has hampered the development of effective multilingual approaches to WSD, recently this idea has been revamped with the organization of SemEval tasks on cross-lingual WSD (Lefever and Hoste, 2010) and cross-lingual lexical substitution (Mihalcea et al., 2010). In addition, new research on the topic has explored the translation of sentences into many languages (Navigli and Ponzetto, 2010; Lefever et al., 2011; Banea and Mihalcea, 2011), as well as the projection of monolingual knowledge onto another language (Khapra et al., 2011).

In our research we focus on knowledge-based methods and tools for multilingual WSD, since knowledge-rich WSD has been shown to achieve high performance across domains (Agirre et al., 2009; Navigli et al., 2011) and to compete with supervised methods on a variety of lexical disambiguation tasks (Ponzetto and Navigli, 2010). Our vision of knowledge-rich multilingual WSD requires two fundamental components: first, a wide-coverage multilingual lexical knowledge base; second, tools to effectively query, retrieve and exploit its information for disambiguation. Nevertheless, to date, no integrated resources and tools exist that are freely available to the research community on a multilingual scale. Previous endeavors are either not freely available (EuroWordNet (Vossen, 1998)), or are only accessible via a Web interface (cf. the Multilingual Research Repository (Atserias et al., 2004) and MENTA (de Melo and Weikum, 2010)), thus providing no programmatic access. And this is despite the fact that the availability of easy-to-use libraries for efficient information access is known to foster top-level research – cf. the widespread use of semantic similarity measures in NLP, thanks to the availability of `WordNet::Similarity` (Pedersen et al., 2004).

With the present contribution we aim to fill this gap in multilingual tools, providing a multi-tiered contribution consisting of (a) an Application Programming Interface (API) for efficiently accessing the information available in BabelNet (Navigli and

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 67–72,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

```
bn:00008364n WIKIWN 08420278n 85 WN:EN:bank WIKI:EN:Bank WIKI:DE:Bank WIKI:IT:Banca
                                 WIKIRED:DE:Finanzinstitut WN:EN:banking_company
                                 WNTR:ES:banco WNTR:FR:société_bancaire WIKI:FR:Banque ...
                              35 1_7 2_3,4,9 6_8 ...
                             228 r bn:02945246n r bn:02854884n|FROM_IT @ bn:00034537n ...
```

Figure 1: The Babel synset for bank$_n^2$, i.e. its 'financial' sense (excerpt, formatted for ease of readability).

Ponzetto, 2010), a very large knowledge repository with concept lexicalizations in 6 languages (Catalan, English, French, German, Italian and Spanish), at the lexicographic (i.e., word senses), encyclopedic (i.e., named entities) and conceptual (i.e., concepts and semantic relations) levels; (b) an API to perform graph-based WSD with BabelNet, thus providing, for the first time, a freely-available toolkit for performing knowledge-based WSD in a multilingual and cross-lingual setting.

2 BabelNet

BabelNet follows the structure of a traditional lexical knowledge base and accordingly consists of a labeled directed graph where nodes represent concepts and named entities and edges express semantic relations between them. Concepts and relations are harvested from the largest available semantic lexicon of English, i.e., WordNet (Fellbaum, 1998), and a wide-coverage collaboratively-edited encyclopedia, i.e., Wikipedia[1], thus making BabelNet a multilingual 'encyclopedic dictionary' which automatically integrates fine-grained lexicographic information with large amounts of encyclopedic knowledge by means of a high-performing mapping algorithm (Navigli and Ponzetto, 2010). In addition to this conceptual backbone, BabelNet provides a multilingual lexical dimension. Each of its nodes, called *Babel synsets*, contains a set of lexicalizations of the concept for different languages, e.g., { bank$_{\text{EN}}$, Bank$_{\text{DE}}$, banca$_{\text{IT}}$, . . . , banco$_{\text{ES}}$ }.

Similar in spirit to WordNet, BabelNet consists, at its lowest level, of a plain text file. An excerpt of the entry for the Babel synset containing bank$_n^2$ is shown in Figure 1[2]. The record contains (a) the synset's id; (b) the region of BabelNet where it lies (e.g., WIKIWN means at the intersection of WordNet and Wikipedia); (c) the corresponding (possibly empty) WordNet 3.0 synset offset; (d) the number of senses in all languages and their full listing; (e) the number of translation relations and their full listing; (f) the number of semantic pointers (i.e., relations) to other Babel synsets and their full listing. Senses encode information about their source – i.e., whether they come from WordNet (WN), Wikipedia pages (WIKI) or their redirections (WIKIRED), or are automatic translations (WNTR / WIKITR) – and about their language and lemma. In addition, translation relations among lexical items are represented as a mapping from source to target senses – e.g., 2_3,4,9 means that the second element in the list of senses (the English word bank) translates into items #3 (German Bank), #4 (Italian banca), and #9 (French banque). Finally, semantic relations are encoded using WordNet's pointers and an additional symbol for Wikipedia relations (r), which can also specify the source of the relation (e.g., FROM_IT means that the relation was harvested from the Italian Wikipedia). In Figure 1, the Babel synset inherits the WordNet hypernym (@) relation to financial institution$_n^1$ (offset bn:00034537n), as well as Wikipedia relations to the synsets of FINANCIAL INSTRUMENT (bn:02945246n) and ETHICAL BANKING (bn:02854884n, from Italian).

3 An API for multilingual WSD

BabelNet API. BabelNet can be effectively accessed and automatically embedded within applications by means of a programmatic access. In order to achieve this, we developed a Java API, based on Apache Lucene[3], which indexes the BabelNet textual dump and includes a variety of methods to access the four main levels of information encoded in BabelNet, namely: (a) lexicographic (information about word senses), (b) encyclopedic (i.e. named en-

[1] http://www.wikipedia.org

[2] We denote with w_p^i the i-th WordNet sense of a word w with part of speech p.

[3] http://lucene.apache.org

```
1  BabelNet bn = BabelNet.getInstance();
2  System.out.println("SYNSETS WITH English word: \"bank\"");
3  List<BabelSynset> synsets = bn.getSynsets(Language.EN, "bank");
4  for (BabelSynset synset : synsets) {
5    System.out.print("  =>(" + synset.getId() + ") SOURCE: " + synset.getSource() +
6                     "; WN SYNSET: " + synset.getWordNetOffsets() + ";\n" +
7                     "  MAIN LEMMA: " + synset.getMainLemma() + ";\n  SENSES (IT): { ");
8    for (BabelSense sense : synset.getSenses(Language.IT))
9      System.out.print(sense.toString()+" ");
10   System.out.println("}\n  -----");
11   Map<IPointer, List<BabelSynset>> relatedSynsets = synset.getRelatedMap();
12   for (IPointer relationType : relatedSynsets.keySet()) {
13     List<BabelSynset> relationSynsets = relatedSynsets.get(relationType);
14     for (BabelSynset relationSynset : relationSynsets) {
15       System.out.println("    EDGE " + relationType.getSymbol() +
16                          " " + relationSynset.getId() +
17                          " " + relationSynset.toString(Language.EN));
18     }
19   }
20   System.out.println("  -----");
21 }
```

Figure 2: Sample BabelNet API usage.

tities), (c) conceptual (the semantic network made up of its concepts), (d) and multilingual level (information about word translations). Figure 2 shows a usage example of the BabelNet API. In the code snippet we start by querying the Babel synsets for the English word bank (line 3). Next, we access different kinds of information for each synset: first, we print their id, source (WordNet, Wikipedia, or both), the corresponding, possibly empty, WordNet offsets, and 'main lemma' – namely, a compact string representation of the Babel synset consisting of its corresponding WordNet synset in stringified form, or the first non-redirection Wikipedia page found in it (lines 5–7). Then, we access and print the Italian word senses they contain (lines 8–10), and finally the synsets they are related to (lines 11–19). Thanks to carefully designed Java classes, we are able to accomplish all of this in about 20 lines of code.

Multilingual WSD API. We use the BabelNet API as a framework to build a toolkit that allows the user to perform multilingual graph-based lexical disambiguation – namely, to identify the most suitable meanings of the input words on the basis of the semantic connections found in the lexical knowledge base, along the lines of Navigli and Lapata (2010). At its core, the API leverages an in-house Java library to query paths and create semantic graphs with BabelNet. The latter works by pre-computing off-line paths connecting any pair of Babel synsets, which are collected by iterating through each synset in turn, and performing a depth-first search up to a maximum depth – which we set to 3, on the basis of experimental evidence from a variety of knowledge base linking and lexical disambiguation tasks (Navigli and Lapata, 2010; Ponzetto and Navigli, 2010). Next, these paths are stored within a Lucene index, which ensures efficient lookups for querying those paths starting and ending in a specific synset. Given a set of words as input, a semantic graph factory class searches for their meanings within BabelNet, looks for their connecting paths, and merges such paths within a single graph. Optionally, the paths making up the graph can be filtered – e.g., it is possible to remove loops, weighted edges below a certain threshold, etc. – and the graph nodes can be scored using a variety of methods – such as, for instance, their outdegree or PageRank value in the context of the semantic graph. These graph connectivity measures can be used to rank senses of the input words, thus performing graph-based WSD on the basis of the structure of the underlying knowledge base.

We show in Figure 3 a usage example of our disambiguation API. The method which performs WSD (`disambiguate`) takes as input a collection of words (i.e., typically a sentence), a `KnowledgeBase` with which to perform dis-

```
1  public static void disambiguate(Collection<Word> words,
2                                  KnowledgeBase kb, KnowledgeGraphScorer scorer) {
3    KnowledgeGraphFactory factory = KnowledgeGraphFactory.getInstance(kb);
4    KnowledgeGraph kGraph = factory.getKnowledgeGraph(words);
5    Map<String, Double> scores = scorer.score(kGraph);
6    for (String concept : scores.keySet()) {
7      double score = scores.get(concept);
8      for (Word word : kGraph.wordsForConcept(concept))
9        word.addLabel(concept, score);
10   }
11   for (Word word : words) {
12     System.out.println("\n\t" + word.getWord() + " -- ID " + word.getId() +
13                        " => SENSE DISTRIBUTION: ");
14     for (ScoredItem<String> label : word.getLabels()) {
15       System.out.println("\t  [" + label.getItem() + "]:" +
16                          Strings.format(label.getScore()));
17     }
18   }
19 }
20
21 public static void main(String[] args) {
22   List<Word> sentence = Arrays.asList(
23     new Word[]{new Word("bank", ’n’, Language.EN), new Word("bonus", ’n’, Language.EN),
24                new Word("pay", ’v’, Language.EN), new Word("stock", ’n’, Language.EN)});
25   disambiguate(sentence, KnowledgeBase.BABELNET, KnowledgeGraphScorer.DEGREE);
26 }
```

Figure 3: Sample Word Sense Disambiguation API usage.

ambiguation, and a `KnowledgeGraphScorer`, namely a value from an enumeration of different graph connectivity measures (e.g., node outdegree), which are responsible for scoring nodes (i.e., concepts) in the graph. `KnowledgeBase` is an enumeration of supported knowledge bases: currently, it includes BabelNet, as well as WordNet++ (namely, an English WordNet-based subset of it (Ponzetto and Navigli, 2010)) and WordNet. Note that, while BabelNet is presently the only lexical knowledge base which allows for multilingual processing, our framework can easily be extended to work with other existing lexical knowledge resources, provided they can be wrapped around Java classes and implement interface methods for querying senses, concepts, and their semantic relations. In the snippet we start in line 3 by obtaining an instance of the factory class which creates the semantic graphs for a given knowledge base. Next, we use this factory to create the graph for the input words (line 4). We then score the senses of the input words occurring within this graph (line 5–10). Finally, we output the sense distributions of each word in lines 11–18. The disambiguation method, in turn, can be called by any other Java program in a way similar to the one highlighted by the `main` method of lines 21–26, where we disambiguate the sample sentence ‘*bank bonuses are paid in stocks*’ (note that each input word can be written in any of the 6 languages, i.e. we could mix languages).

4 Experiments

We benchmark our API by performing knowledge-based WSD with BabelNet on standard SemEval datasets, namely the SemEval-2007 coarse-grained all-words (Navigli et al., 2007, Coarse-WSD, henceforth) and the SemEval-2010 cross-lingual (Lefever and Hoste, 2010, CL-WSD) WSD tasks. For both experimental settings we use a standard graph-based algorithm, Degree (Navigli and Lapata, 2010), which has been previously shown to yield a highly competitive performance on different lexical disambiguation tasks (Ponzetto and Navigli, 2010). Given a semantic graph for the input context, Degree selects the sense of the target word with the highest vertex degree. In addition, in the CL-WSD setting we need to output appropriate lexicalization(s) in different languages. Since the selected Babel synset can contain multiple translations in a target language for the given English word, we use for this task an

Algorithm	Nouns only	All words
NUS-PT	82.3	82.5
SUSSX-FR	81.1	77.0
Degree	84.7	82.3
MFS BL	77.4	78.9
Random BL	63.5	62.7

Table 1: Performance on SemEval-2007 coarse-grained all-words WSD (Navigli et al., 2007).

unsupervised approach where we return for each test instance only the most frequent translation found in the synset, as given by its frequency of alignment obtained from the Europarl corpus (Koehn, 2005).

Tables 1 and 2 summarize our results in terms of recall (the primary metric for WSD tasks): for each SemEval task, we benchmark our disambiguation API against the best unsupervised and supervised systems, namely SUSSX-FR (Koeling and McCarthy, 2007) and NUS-PT (Chan et al., 2007) for Coarse-WSD, and T3-COLEUR (Guo and Diab, 2010) and UvT-v (van Gompel, 2010) for CL-WSD. In the Coarse-WSD task our API achieves the best overall performance on the nouns-only subset of the data, thus supporting previous findings indicating the benefits of using rich knowledge bases like BabelNet. In the CL-WSD evaluation, instead, using BabelNet allows us to surpass the best unsupervised system by a substantial margin, thus indicating the viability of high-performing WSD with a multilingual lexical knowledge base. While our performance still lags behind the application of supervised techniques to this task (cf. also results from Lefever and Hoste (2010)), we argue that further improvements can still be obtained by exploiting more complex disambiguation strategies. In general, using our toolkit we are able to achieve a performance which is competitive with the state of the art for these tasks, thus supporting previous findings on knowledge-rich WSD, and confirming the robustness of our toolkit.

5 Related Work

Our work complements recent efforts focused on visual browsing of wide-coverage knowledge bases (Tylenda et al., 2011; Navigli and Ponzetto, 2012) by means of an API which allows the user to programmatically query and search BabelNet. This knowledge resource, in turn, can be used for easily performing multilingual and cross-lingual WSD out-of-the-box. In comparison with other contributions, our toolkit for multilingual WSD takes previous work from Navigli (2006), in which an online interface for graph-based monolingual WSD is presented, one step further by adding a multilingual dimension as well as a full-fledged API. Our work also complements previous attempts by NLP researchers to provide the community with freely available tools to perform state-of-the-art WSD using WordNet-based measures of semantic relatedness (Patwardhan et al., 2005), as well as supervised WSD techniques (Zhong and Ng, 2010). We achieve this by building upon BabelNet, a multilingual 'encyclopedic dictionary' bringing together the lexicographic and encyclopedic knowledge from WordNet and Wikipedia. Other recent projects on creating multilingual knowledge bases from Wikipedia include WikiNet (Nastase et al., 2010) and MENTA (de Melo and Weikum, 2010): both these resources offer structured information complementary to BabelNet – i.e., large amounts of facts about entities (MENTA), and explicit semantic relations harvested from Wikipedia categories (WikiNet).

	Degree	T3-Coleur	UvT-v
Dutch	15.52	10.56	17.70
French	22.94	21.75	–
German	17.15	13.05	–
Italian	18.03	14.67	–
Spanish	22.48	19.64	23.39

Table 2: Performance on SemEval-2010 cross-lingual WSD (Lefever and Hoste, 2010).

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

BabelNet and its API are available for download at `http://lcl.uniroma1.it/babelnet`.

References

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2009. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In *Proc. of IJCAI-09*, pages 1501–1506.

Jordi Atserias, Luis Villarejo, German Rigau, Eneko Agirre, John Carroll, Bernardo Magnini, and Piek Vossen. 2004. The MEANING multilingual central repository. In *Proc. of GWC-04*, pages 22–31.

Carmen Banea and Rada Mihalcea. 2011. Word Sense Disambiguation with multilingual features. In *Proc. of IWCS-11*, pages 25–34.

Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. 2007. NUS-ML: Exploiting parallel texts for Word Sense Disambiguation in the English all-words tasks. In *Proc. of SemEval-2007*, pages 253–256.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proc. of ACL-11*, pages 600–609.

Gerard de Melo and Gerhard Weikum. 2010. MENTA: inducing multilingual taxonomies from Wikipedia. In *Proc. of CIKM-10*, pages 1099–1108.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Weiwei Guo and Mona Diab. 2010. COLEPL and COLSLM: An unsupervised WSD approach to multilingual lexical substitution, tasks 2 and 3 SemEval 2010. In *Proc. of SemEval-2010*, pages 129–133.

Mitesh M. Khapra, Salil Joshi, Arindam Chatterjee, and Pushpak Bhattacharyya. 2011. Together we can: Bilingual bootstrapping for WSD. In *Proc. of ACL-11*, pages 561–569.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*.

Rob Koeling and Diana McCarthy. 2007. Sussx: WSD using automatically acquired predominant senses. In *Proc. of SemEval-2007*, pages 314–317.

Els Lefever and Veronique Hoste. 2010. SemEval-2010 Task 3: Cross-lingual Word Sense Disambiguation. In *Proc. of SemEval-2010*, pages 15–20.

Els Lefever, Véronique Hoste, and Martine De Cock. 2011. Parasense or how to use parallel corpora for Word Sense Disambiguation. In *Proc. of ACL-11*, pages 317–322.

Bin Lu, Chenhao Tan, Claire Cardie, and Benjamin K. Tsou. 2011. Joint bilingual sentiment classification with unlabeled parallel corpora. In *Proc. of ACL-11*, pages 320–330.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *Proc. of ACL-11*, pages 1336–1345.

Rada Mihalcea, Ravi Sinha, and Diana McCarthy. 2010. SemEval-2010 Task 2: Cross-lingual lexical substitution. In *Proc. of SemEval-2010*, pages 9–14.

Vivi Nastase, Michael Strube, Benjamin Börschinger, Caecilia Zirn, and Anas Elghafari. 2010. WikiNet: A very large scale multi-lingual concept network. In *Proc. of LREC '10*.

Roberto Navigli and Mirella Lapata. 2010. An experimental study on graph connectivity for unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proc. of ACL-10*, pages 216–225.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNetXplorer: a platform for multilingual lexical knowledge base access and exploration. In *Comp. Vol. to Proc. of WWW-12*, pages 393–396.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proc. of SemEval-2007*, pages 30–35.

Roberto Navigli, Stefano Faralli, Aitor Soroa, Oier Lopez de Lacalle, and Eneko Agirre. 2011. Two birds with one stone: learning semantic models for Text Categorization and Word Sense Disambiguation. In *Proc. of CIKM-11*, pages 2317–2320.

Roberto Navigli. 2006. Online word sense disambiguation with structural semantic interconnections. In *Proc. of EACL-06*, pages 107–110.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2005. SenseRelate::TargetWord – a generalized framework for Word Sense Disambiguation. In *Comp. Vol. to Proc. of ACL-05*, pages 73–76.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – Measuring the relatedness of concepts. In *Comp. Vol. to Proc. of HLT-NAACL-04*, pages 267–270.

Yves Peirsman and Sebastian Padó. 2010. Cross-lingual induction of selectional preferences with bilingual vector spaces. In *Proc. of NAACL-HLT-10*, pages 921–929.

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich Word Sense Disambiguation rivaling supervised system. In *Proc. of ACL-10*, pages 1522–1531.

Tomasz Tylenda, Mauro Sozio, and Gerhard Weikum. 2011. Einstein: physicist or vegetarian? Summarizing semantic type graphs for knowledge discovery. In *Proc. of WWW-11*, pages 273–276.

Maarten van Gompel. 2010. UvT-WSD1: A cross-lingual word sense disambiguation system. In *Proc. of SemEval-2010*, pages 238–241.

Piek Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer, Dordrecht, The Netherlands.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proc. of ACL-10 System Demonstrations*, pages 78–83.