Semantic services in FreeLing 2.1: WordNet and UKB

Lluís Padró
Software Department
TALP Research Center
Universitat Politècnica de Catalunya
padro@lsi.upc.edu

Samuel Reese
ISAE - Supaero
Université Paul Sabatier
samuel.reese@supaero.org

Eneko Agirre, Aitor Soroa
IXA NLP Group
University of the Basque Country
{e.agirre,a.soroa}@ehu.es

Abstract

FreeLing is an open-source open-source multilingual language processing library providing a wide range of language analyzers for several languages. It offers text processing and language annotation facilities to natural language processing application developers, simplifying the task of building those applications. FreeLing is customizable and extensible. Developers can use the default linguistic resources (dictionaries, lexicons, grammars, etc.) directly, or extend them, adapt them to specific domains, or even develop new ones for specific languages.

This paper presents the semantic services included in FreeLing, which are based on WordNet and EuroWordNet databases. The recent release of the UKB program under a GPL license made it possible to integrate a long awaited word sense disambiguation module into FreeLing. UKB provides state of the art all-words sense disambiguation for any language with an available WordNet.

1 Introduction

Basic language processing tasks such as tokenizing, morphological analysis, lemmatizing, part-of-speech tagging, word sense disambiguation (WSD), dependency parsing, etc. are needed for most natural language processing (NLP) applications such as Machine Translation, Summarization, Dialogue systems, Text mining, etc.

This makes language analyzers a very valuable resources for researchers and developers in NLP. Also, the lack of out-of-the-box state-of-the-art systems is a severe bottleneck for faster progress in the area, both in research and development.

Additionally, a large part of the effort required to develop NLP systems is devoted to the adaptation of existing software resources to the platform, I/O format, or API of the final application.

FreeLing was undertaken with the belief that steps should be taken towards general availability of basic NLP tools and resources, which may be used without restrictions. Thus, to enable faster advances and more portable systems in our area, an open–source model was chosen.

After five years (first version was released on 2004), over 10,000 downloads, and a growing user community which has extended the initial three languages (English, Spanish and Catalan) to seven (adding Galician, Italian, Welsh, Portuguese, and Asturian) prove that the collaborative open model is a productive approach to the development of NLP tools and resources.

In this paper, we focus on the FreeLing services related to semantic processing, namely wordnet access and word sense disambiguation. The next section presents the internal structure of the library. Sections 3 and 4 present the wordnet access and WSD services. Section 5 depicts some examples, and Section 6 outlines some conclusions.

2 Data structure and language analysis services

FreeLing is conceived as a library on top of which powerful NLP applications can be developed, and oriented to ease the integration of language analysis services into higher level applications.

Its architecture consists of a simple two-layer client-server approach: A basic linguistic service layer which provides analysis services (morphological analysis, tagging, parsing, ...), and an application layer which, acting as a client, requests the desired services from the analyzers.

The library is written in C++, since speed is a must for real-world oriented applications. Additionaly, APIs are provided to call the library services from Java, perl, and pyhton.

The internal architecture of the system is based on two kinds of objects: linguistic data objects and processing objects.

2.1 Linguistic Data Classes

The basic classes in the library are used to contain linguistic data (such as a word, a PoS tag, a sentence, a document...). Any client application must be aware of those classes in order to be able to provide to each processing module the right data, and to correctly interpret the module results.

The linguistic classes supported by the current version are:

- `analysis`: A tuple <lemma, PoS tag, probability, sense list>.
- `word`: A word form with a list of possible `analysis` objects.
- `sentence`: A list of `word` known to be a complete sentence, it may include also a parse tree and/or a dependency tree.
- `paragraph`: A list of `sentence` known to be an independent paragraph.
- `document`: A list of `paragraph` that form a complete document. It may contain also coreference information about the entity mentions in the document.

Figure 1 presents a UML diagram with the linguistic data classes.

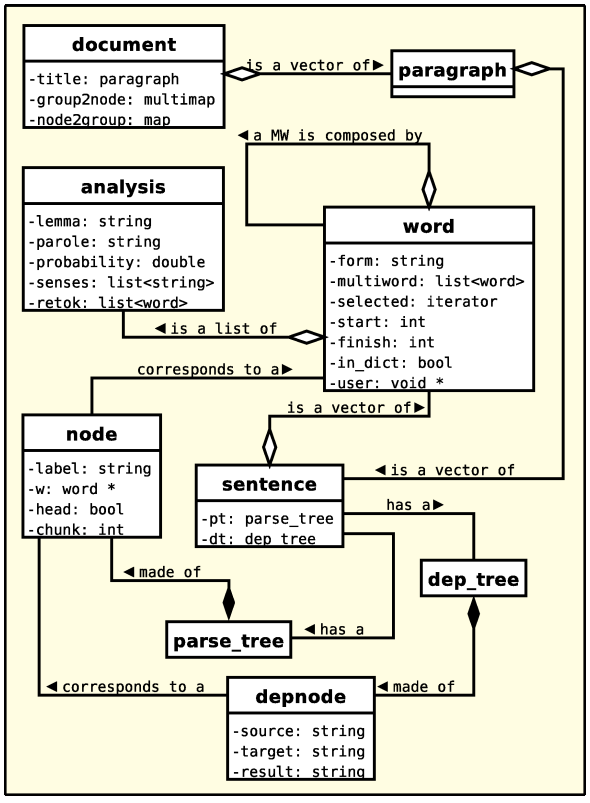

Figure 1: FreeLing-2.1 Linguistic Data Classes.

2.2 Processing Classes

Apart from classes containing linguistic data, the library provides classes able to transform them. See Figure 2 below for a UML diagram.

- `tokenizer`: Receives plain text and returns a list of `word` objects.
- `splitter`: Receives a list of `word` objects and returns a list of `sentence` objects.
- `morfo`: Receives a list of `sentence` and morphologically annotates each `word` of each sentence in the list. In fact, this class applies a cascade of specialized processors (number detection, date/time detection, multi-word detection, dictionary search, etc.) each of which is in turn a processing class:
  - `locutions`: Multi-word recognizer.
  - `dictionary`: Dictionary lookup and suffix handling.
  - `numbers`: Numerical expressions recognizer.
  - `dates`: Date/time expressions recognizer.
  - `quantities`: Ratio and percentage expressions and monetary amount recognizer.
  - `punts`: Punctuation symbol annotator.
  - `probabilities`: Lexical probabilities annotator and unknown word handler.
  - `np`: Proper noun recognizer.
- `tagger`: Receives a list of `sentence` and disambiguates the PoS of each `word` in the given sentences. If the selected analysis carries retokenization information, the word may be split in two or more new words.
- NE `classifier`: Receives a list of `sentence` and classifies all `word` tagged as proper nouns in the given sentences.
- `Sense annotator`: Receives a list of `sentence` and adds synset information to the selected `analysis` for each `word`.
- `Word sense disambiguator`: Receives a list of `sentence` and ranks the possible senses for each `word` selected `analysis`.
- `chunk parser`: Receives a list of `sentence` and enriches each of them with a `parse_tree`.
- `dependency parser`: Receives a list of parsed `sentence` and enriches each of them with a `dependency_tree`.
- `coreference solver`: Receives a document formed by parsed `sentence` and enriches the document with coreference information.

3 Semantic services: WordNet access

There are two basic semantic services: First, a basic database access module that enables the client application to consult a WordNet (Miller et al., 1991) structure (e.g. to find out which synsets a lemma belongs to, which words are contained in one synset, or which are the hypernyms of certain synset). Second, a knowledge-based word sense disambiguator, which has been recently integrated thanks to the release of UKB disambiguator under a GPL license (Agirre and Soroa, 2009).

3.1 SemanticDB module

This module handles WordNet-like structures, which are indexed in a local database. The database sources are provided with FreeLing, and can be adapted –or completely changed– to match the application needs.

The source database consists of two files:

- The WN structure file contains a list of synset codes, with information about its PoS, its hypernyms, its WN semantic file, and its features in EuroWordNet TCO (Álvez et al., 2008). For instance, the entry in this file for WN1.6 noun synset {`01630731 cat,true_cat`} is:

  ```
  01630731:N 01630126 05 Animal:Object
  ```

  This file is indexed and used to find out synset properties or their hypernyms.

- The language lexical file contains direct and inverse links between lemmas and synset codes. For instance, the first line in the example below establishes a link from the noun lemma *cat* to all the synsets it belongs to. If they are provided sorted by frequency, the first one can be used to perform most-frequent-sense disambiguation. The two last lines define which words are contained in the given synsets:

  ```
  W:cat:N 01630731 07306044 07143161
  S:01630731:N cat true_cat
  S:07306044:N cat guy hombre
  ```

Note that this module does not (yet) offer as advanced functionalities as the standard WordNet search library, but it has the following advantages:

- Source files are plain text and easy to build. Indexing programs are provided with FreeLing to enable anyone to create his/her own semantic database.
- Language and WN structure files are separated, making it possible to use the structure file as an ILI and map all languages to the same structure if necessary.
- The synset codes serve as mere concept identifiers, so they can be replaced by any other semantic code (e.g. later WN versions synset codes, or even ad-hoc concept codes).
- Being open-source, the capabilities of the module can be easily extended (e.g. to include more semantic tags or more relations in the structure file), or customized to one specific needs.

Currently, FreeLing includes only semantic data for English, Spanish, and Catalan, that are the only languages that offer a version in the Global WordNet Grid under an open-source license.

3.1.1 Use of semantic information by FreeLing modules

The Semantic DB module can be used directly by the client application, but it is also used by other modules in FreeLing:

- The sense annotator: Accesses the database and enriches the text with all possible synsets for each form.
- The relaxation–labelling tagger (Padró, 1998): Deals with constraint-grammar-like rules dealing with PoS tag, form, lemma, or sense to guide the selection of the right analysis.
- The dependency parser (Atserias et al., 2005; Carrera et al., 2008): Uses heuristic rules dealing with PoS, syntax, senses, and TCO information to combine into a complete dependency tree the chunks produced by the shallow parser.
- The coreference solver –based on (Soon et al., 2001): Uses TCO and hypernym relations between two mentions as features used by a machine learning classifier to determine whether they corefer.

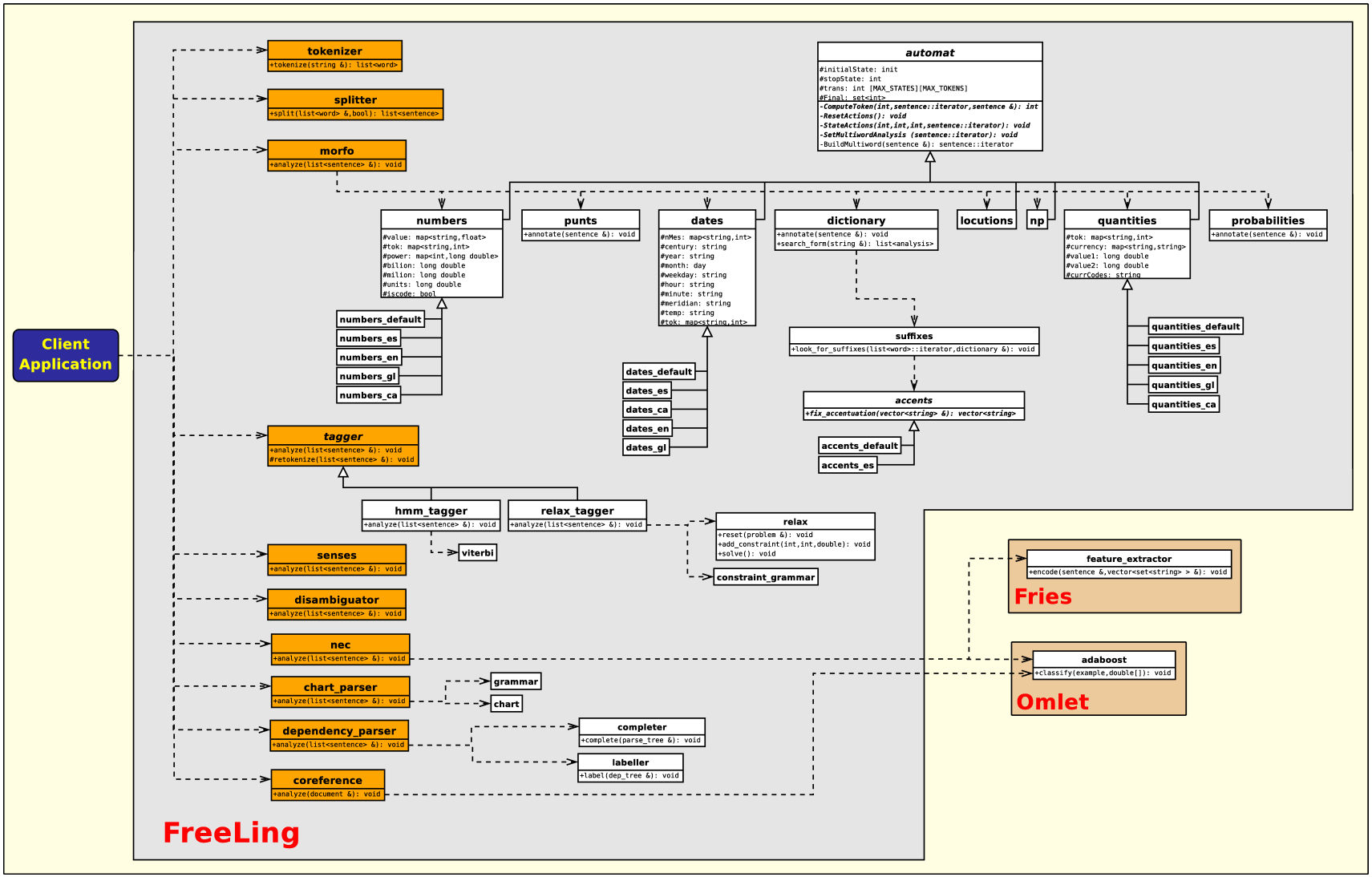

Figure 2: FreeLing-2.1 Main Processing Classes.

4 Semantic services: UKB word sense disambiguation

The PageRank-based word sense disambiguation algorithm UKB (Agirre and Soroa, 2009), and the availability of its code under GPL has recently made it possible to include a long-awaited feature in FreeLing: A language-independent state-of-the-art all-words WSD module. UKB uses the structure of local wordnets in order to perform WSD, and it can be easily applied to any language, with the only requirement of having a wordnet.

The original code has been integrated *as is*, and a simple wrapper has been developed that loads the sentences being analyzed by FreeLing into the appropriate UKB data structure (after the lemmatizer and the tagger have chosen the right PoS and lemma for each word), calls the disambiguator, and loads its results back to the FreeLing data structure. In this way, the UKB module enriches the analysis of a set of sentences with the ranked list of synsets for each word.

Knowledge files handled by this module are:

- The dictionary file, which contains the association between words and synset codes. The same file described above used by the semantic DB module is used. It is converted to the format needed by UKB at installation time. Conversion programs are provided with FreeLing to enable the user to handle his/her own dictionaries.
- The relation graph, containing all relations between synsets to be used by the PageRank algorithm. Since this file contains relations other than hyper/hyponymy, it is currently provided separately, in text format, and indexed at installation time (indexing programs are also provided). Ideally, in the near future this file and the WN structure file used by the SemanticDB should be unified.

Note that, again, the UKB algorithm is a generic graph-based disambiguation tool, which can be fed with any sense dictionary and any relation graph for those senses. Currently, synset codes and relations from wordnets are used, but this module can be used to disambiguate on any sense repository just changing the used knowledge files.

Since this approach of keeping knowledge/data components as separated as possible from processing/code components is also followed by FreeLing, they match easily and both together form a very flexible and sound platform to develop syntax and semantic analyzers for any language.

5 Examples

In this section we will show some simple examples of the semantic capabilities of FreeLing and its UKB component. An online demo of the whole system can be found at `http://www.lsi.upc.edu/~nlp/freeling`.

5.1 Basic sense annotation

The basic semantic functionality is mere sense annotation, enriching a PoS-tagged sentence with a list of possible senses for each word. An example is shown in Figure 3.

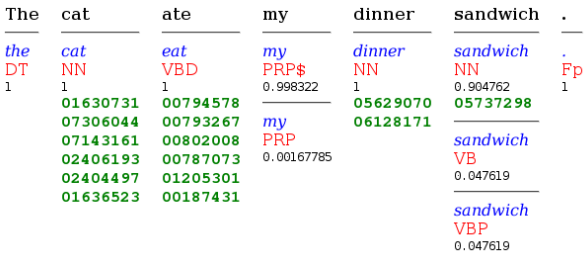

Figure 3: Sense annotation of a PoS-tagged sentence.

If this annotation takes place before PoS tagging, the tagger may use the semantic information to help the disambiguation (e.g. a Constraint Grammar based tagger). If that is not the case, the annotation can take place either before or after the tagging, depending on the user's needs.

If the synset codes provided in the sense dictionary are sorted by frequency, the user application only needs to pick the first one to have a basic MFS disambiguator.

5.2 Semantics used by other FreeLing modules

The module in FreeLing that –currently– takes the larger advantage from the availability of semantic information is the dependency parser. The parser is based on a set of heuristic rules that combine chunks and label their dependencies. See (Atserias et al., 2005; Carrera et al., 2008) for details.

Those heuristic rules may refer to certain properties of the chunks (e.g. head PoS tag, head lemma, position relative to other chunks) including semantic features (TCO properties, WN semantic file, hypernyms).

For instance, consider the Spanish sentences *Juan vió a su amigo* (Juan saw his friend) and *Juan escribió a su amigo* (Juan wrote to his friend). In

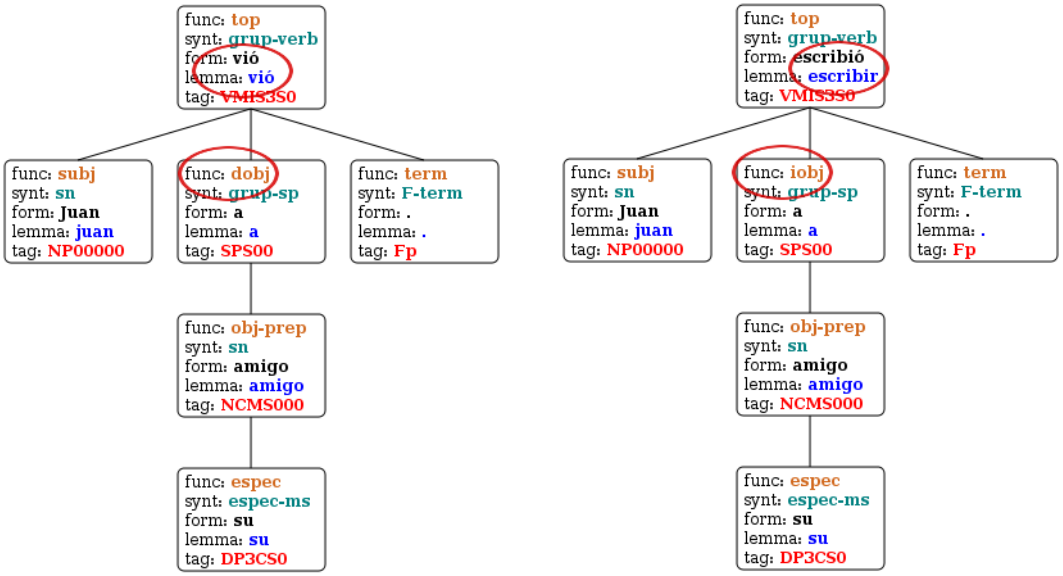

Figure 4: Analysis requiring semantics in dependency parsing

the former, *his friend* is the direct object of the verb *to see*, and in the later, it is the indirect object of *to write*.

The reason is that transitive Spanish verbs such as *to see* that do not have indirect object require the use of the preposition *a* when the direct object is a person. On the other hand, for ditransitive verbs such as *to write* the preposition *a* marks the indirect object.

So, to properly parse these sentences, rules have to be able to check about the `Human` condition of the candidate objects. This is achieved thanks to the TCO access provided by the SemanticDB module, as illustrated in Figure 4.

Another module that benefits from the semantic knowledge included in FreeLing is the machine-learning based coreference solver. The solver considers pairs of nominal mentions (noun phrases and pronouns) and uses a classifier based on (Soon et al., 2001) to determine whether they corefer.

The features used by the classifier include morphosyntax features such as the distance between the mentions, their relative positions, whether they are definite noun phrases, personal pronouns, their gender, number, etc.

They also include semantic information on the kind of entity they may be referring to: If the noun phrase head is a proper noun, a NE classifier is used to determine if it is a person, an organization, or a geographical name. If the noun phrase head is a common noun, its TCO properties are checked to find out whether it is `Human`, `Group` or `Place`. Then, this information is provided as features to the classifier.

5.3 Word Sense Disambiguation

The frequency-ordered semantic dictionaries enable the user to perform a straightforward most-frequent-sense disambiguation just picking the first sense in the list.

The integration of the UKB module (Agirre and Soroa, 2009) offers a more informed disambiguation mechanism. The sense list is ordered according to the PageRank assigned by the algorithm. The user application can simply select the first one, or use the rank information to perform any desired action.

The example sentences in Figure 5 illustrate how UKB is able to distinguish the two main senses for the word *bank* in different contexts, instead of choosing always the same, as a MFS disambiguator would.

Note that this doesn't mean that UKB has a higher accuracy than MFS at WSD. As reported by (Agirre and Soroa, 2009), the results of UKB at the performed experiments on English and Spanish are quite near of MFS results, and clearly improve those of other unsupervised WSD systems.

6 Conclusions

We presented the semantic services included in the FreeLing 2.1 library, which includes access to wordnets and graph-based all-word sense disambiguation on those wordnet, using the state-

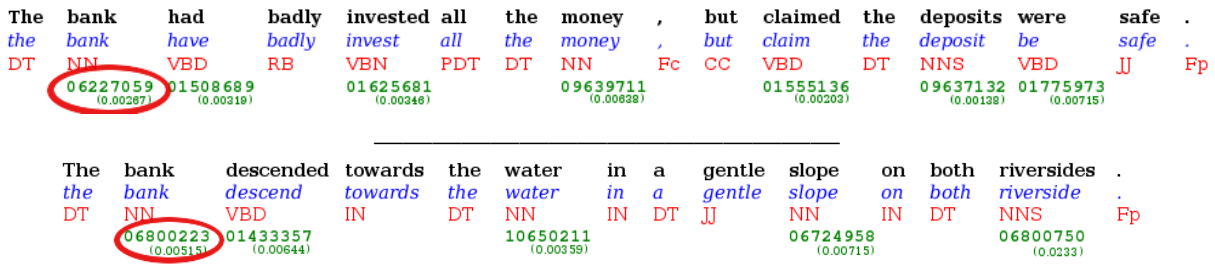

Figure 5: UKB disambiguation of *bank* in different contexts.

of-the-art UKB system (Agirre and Soroa, 2009). The open source licence of these software tools and their architecture, which completely separates code from linguistic data, makes it possible to easily adapt them to any domain or application needs, and provides a platform for affordable development of analyzers for new languages.

Acknowledgements

This work has been partially funded by the Spanish Science and Innovation Ministry, via the KNOW project (TIN2006-15049-C3-03). For further details visit `http://ixa.si.ehu.es/know`.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009)*, Athens, Greece.

Jordi Atserias, Elisabet Comelles, and Aingeru Mayor. 2005. Txala un analizador libre de dependencias para el castellano. *Procesamiento del Lenguaje Natural*, (35):455–456, September.

Jordi Carrera, Irene Castellón, Marina Lloberes, Lluís Padró, and Nevena Tinkova. 2008. Dependency grammars in freeling. *Procesamiento del Lenguaje Natural*, (41):21–28, September.

G. A. Miller, R. Beckwith, Christiane Fellbaum, D. Gross, K. Miller, and R. Tengi. 1991. Five papers on wordnet. *Special Issue of the International Journal of Lexicography*, 3(4):235–312.

Lluís Padró. 1998. *A Hybrid Environment for Syntax–Semantic Tagging*. Ph.D. thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya, February. http://www.lsi.upc.es/˜padro.

W.M. Soon, H. T. Ng, and D.C.Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Javier Álvez, Jordi Atserias, Jordi Carrera, Salvador Climent, Egoitz Laparra, Antoni Oliver, and German Rigau. 2008. Complete and consistent annotation of wordnet using the top concept ontology. In *Proceedings of the the 6th Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech (Morocco), May.