FreeLing 2.1: Five years of open-source language processing tools

Lluís Padró[*], Miquel Collado[*], Samuel Reese[◇], Marina Lloberes[†], Irene Castellón[†]

[*]Software Department – TALP Research Center
Universitat Politècnica de Catalunya
padro@lsi.upc.edu, miquel.collado@gmail.com

[◇]ISAE – Supaero
Université Paul Sabatier
samuel.reese@supaero.org

[†]GRIAL research group
Universitat de Barcelona
{marina.lloberes,icastellon}@ub.edu

Abstract

FreeLing is an 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 overviews the recent history of this tool, summarizes the improvements and extensions incorporated in the latest version, and depicts the architecture of the library. Special focus is brought to the fact and consequences of the library being open-source: After five years and over 35,000 downloads, a growing user community has extended the initial three languages (English, Spanish and Catalan) to eight (adding Galician, Italian, Welsh, Portuguese, and Asturian), proving that the collaborative open model is a productive approach for the development of NLP tools and resources.

1. Introduction

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

This makes language analyzers 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.

FreeLing was undertaken with the believe 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 35,000 downloads, and a growing user community which has extended the initial three languages (English, Spanish and Catalan) to eight (adding Galician, Italian, Welsh, Portuguese, and Asturian) prove that the collaborative open model is a productive approach for the development of NLP tools and resources.

In the next section we review FreeLing evolution in its five-year life. Section 3 describes the most recent enhancements to the library, and Section 4 presents a shallow technical overview of the library. Section 5 outlines some conclusions and further work.

2. FreeLing project evolution

The first version of FreeLing (1.0) was released in 2003, and presented to the community in LREC 2004 (Carreras et al., 2004). That first version was a reimplementation of several tools (Atserias et al., 1998; Carreras and Padró, 2002) developed at the TALP research center[1] in UPC[2].

The new version, was rewritten from scratch with the following goals:

- Build a fast state-of-the-art analyzer that could be used to process large amounts of text and to develop NLP applications.

[1] http://www.talp.upc.edu

[2] http://www.upc.edu

- Disseminate the results of the NLP research at TALP center, and make it available to the community
- Set a collaborative environment for the development of language analyzers, where contributions from any NLP community member can fit and revert in a global benefit.

To achieve the these goals, the tool was structured as a highly modular C++ library consisting of several classes, each able to perform a different kind of analysis. Under this approach, new classes or services can be added to the library, and applications can select which services will be used, how, and when.

The goals related to dissemination and community collaboration were achieved releasing the software under an open-source GPL license, which has made it possible to enrich the project with many contributions, and make the tool useful for a larger number of people, as the increasing number of downloads suggests.

Currently, the number of downloads of the library is over 35,000 since it was first released, showing a steady growth at each release, and a large expansion at the latest 2.1 version (see Table 1).

Versions		total #downl	monthly average #downl
1.0, 1.0.1, 1.1	Oct.03–Sep.04	608	54
1.2, 1.3, 1.4, 1.5b1	Oct.04–Sep.06	2,250	96
1.5	Oct.06–Jan.08	2,451	161
2.0, 2.1a1, 2.1b1,	Feb.08–Sep.09	3,297	163
2.1a1, 2.1	Oct.09–Mar.10	26,713	4,452

Table 1: Evolution of FreeLing downloads

The original three supported languages (Catalan, Spanish, and English) have been brought up to eight by this community, with the inclusion of Italian, Galician, Portuguese, Welsh, and Asturian. Also, the original size-limited dictionaries for Catalan and Spanish have been replaced by much larger state-of-the art resources.

Finally, the increasing availability of other open-source language analysis tools has enabled us to extend the library capabilities with long-awaited services such as Word Sense Disambiguation (WSD), obtained thanks to the inclusion of the UKB disambiguator developed by (Agirre and Soroa, 2009). See (Padró et al., 2010) for details on its integration in FreeLing.

We regard this evolution as very satisfactory, since it is fulfilling our goals of creating a useful tool which can be enlarged by the community, used by application developers, and become a flexible platform enabling the integration of new resources and services.

3. Recent extensions

The most recent extensions included in FreeLing library can be classified in three types: extensions for linguistic resources for existing languages, support for new languages, and processing modules offering new services.

3.1. Improving existing languages

As in most open-source projects, a large part of FreeLing contributions are related to the enhancement of already existing features. In our case, noticeable contributions include:

- Development of a number recognition module for Italian, which was not present in previous versions, thanks to Vitalie Scurtu[3].
- Enlargement of Spanish dictionary from 6,000 to 75,000 lemmas, thanks to the Spanish Resource Grammar (SRG) project (Marimon et al., 2007).
- Enlargement of the Catalan dictionary from 7,000 to 71,000 lemmas, thanks to the Catalan speller project *El Corrector*[4].
- Enlargement of the Galician dictionary from 7,000 to 50,000 lemmas, thanks to the PLN research group at Universidade de Santiago de Compostela[5].
- Development and improvement of Spanish, Catalan, and English chunking and dependency grammars by the GRIAL research group[6] (Carrera et al., 2008), and thanks to EuroOpenTrad and KNOW projects, funded by Spanish Government

3.2. Support for new languages

Another important bulk of contributions are those devoted to include support for new languages in FreeLing. This task requires on the one hand, the compilation of a morphological dictionary which provides possible lemmas and PoS tags for each form, and on the other hand, the development of a hand tagged corpus (consistent with the dictionary) to enable the training of the PoS taggers. In addition, other minor adjustments have to be done, such as adapting the tokenizer rules to the particularities of the language, writing appropriate rules to deal with suffixation, etc.

Languages that have been recently included are Welsh, Asturian, and Portuguese. All of them offer a chain of processing from plain text up to PoS tagging.

[3] `http://scurtu.sitonline.it`

[4] `http://www.elcorrector.cat`

[5] `http://gramatica.usc.es`

[6] `http://grial.uab.es`

3.3. Modules offering new services

There are several recently added functionalities, some are small (such as the possibility of prefix handling in morphological analysis), some are entire new modules offering a new NLP analysis task, such as Word Sense Disambiguation or Coreference Resolution:

- A language-independent WSD module has been integrated. The code is straightforwardly extracted from the UKB disambiguator[7] developed by (Agirre and Soroa, 2009). The only requirement to use that module on a certain language is the existence of a WordNet-like ontology that relates the words to senses, and the senses among themselves.
- A coreference resolution module has been developed from scratch, using Machine Learning techniques. The approach is based on that of (Soon et al., 2001), and currently, trained models are provided only for Spanish (still with under state-of-the-art accuracy).

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

4.1. 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.

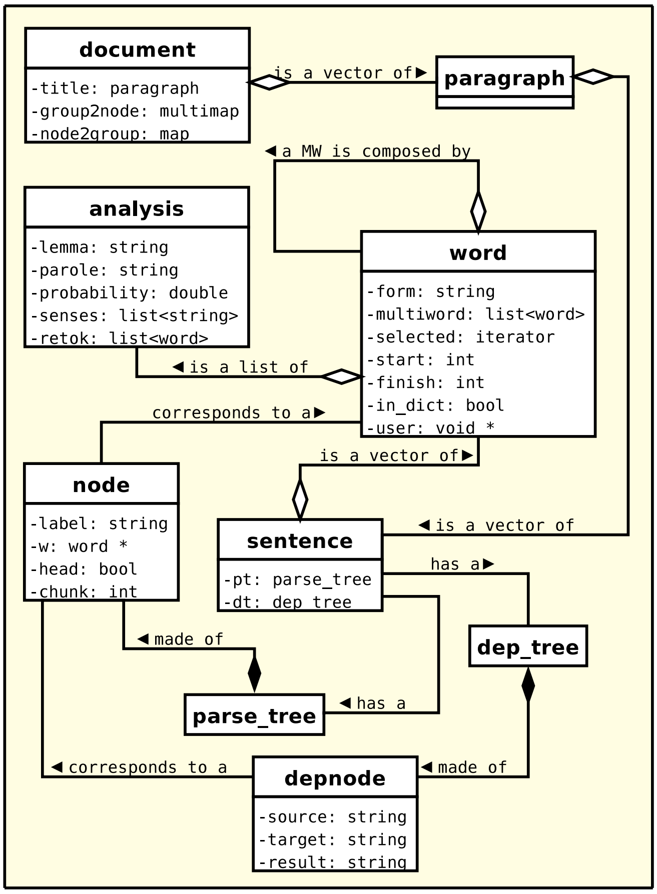

Figure 1: FreeLing-2.1 Linguistic Data Classes.

4.2. Processing Classes

Apart from classes containing linguistic data, the library provides classes able to transform them. A UML diagram can be found in Figure 2.

- `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.

[7] `http://ixa2.si.ehu.es/ukb/`

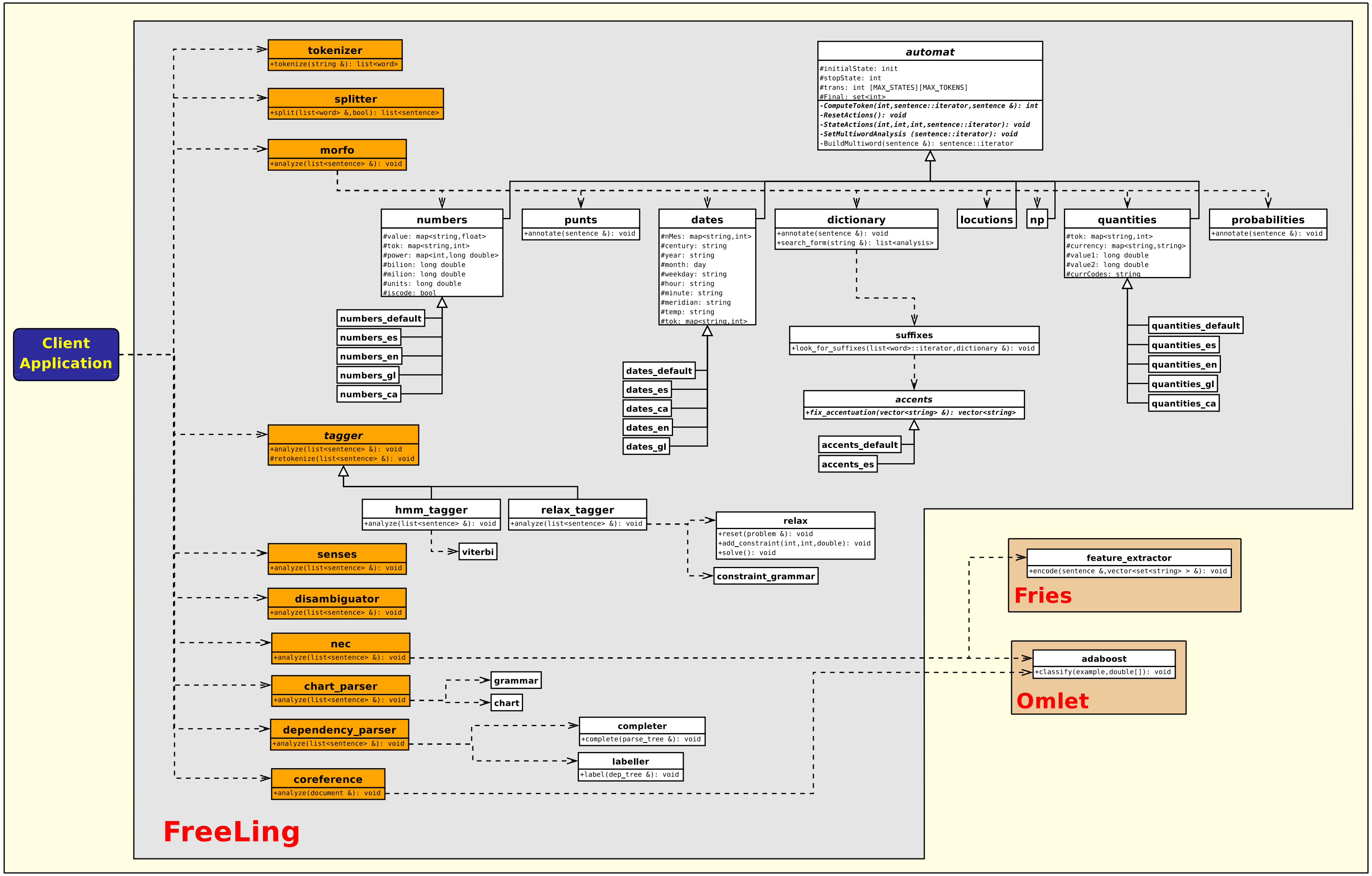
Client Application
tokenizer
+tokenize(string &): list<word>
splitter
+split(list<word> &,bool): list<sentence>
morfo
+analyze(list<sentence> &): void
automat
#initialState: init
#stopState: int
#trans: int [MAX_STATES][MAX_TOKENS]
#Final: set<int>
-ComputeToken(int,sentence::iterator,sentence &): int
-ResetActions(): void
-StateActions(int,int,int,sentence::iterator): void
-SetMultiwordAnalysis (sentence::iterator): void
-BuildMultiword(sentence &): sentence::iterator
numbers
#value: map<string,float>
#tok: map<string,int>
#power: map<int,long double>
#bilion: long double
#milion: long double
#units: long double
#iscode: bool
punts
+annotate(sentence &): void
dates
#nMes: map<string,int>
#century: string
#year: string
#month: day
#weekday: string
#hour: string
#minute: string
#meridian: string
#temp: string
#tok: map<string,int>
dictionary
+annotate(sentence &): void
+search_form(string &): list<analysis>
locutions
np
quantities
#tok: map<string,int>
#currency: map<string,string>
#value1: long double
#value2: long double
#currCodes: string
probabilities
+annotate(sentence &): void
numbers_default
numbers_es
numbers_en
numbers_gl
numbers_ca
dates_default
dates_es
dates_ca
dates_en
dates_gl
suffixes
+look_for_suffixes(list<word>::iterator,dictionary &): void
accents
+fix_accentuation(vector<string> &): vector<string>
accents_default
accents_es
quantities_default
quantities_es
quantities_en
quantities_gl
quantities_ca
tagger
+analyze(list<sentence> &): void
#retokenize(list<sentence> &): void
hmm_tagger
+analyze(list<sentence> &): void
relax_tagger
+analyze(list<sentence> &): void
viterbi
relax
+reset(problem &): void
+add_constraint(int,int,double): void
+solve(): void
constraint_grammar
senses
+analyze(list<sentence> &): void
disambiguator
+analyze(list<sentence> &): void
nec
+analyze(list<sentence> &): void
chart_parser
+analyze(list<sentence> &): void
grammar
chart
dependency_parser
+analyze(list<sentence> &): void
completer
+complete(parse_tree &): void
labeller
+label(dep_tree &): void
coreference
+analyze(document &): void
FreeLing
feature_extractor
+encode(sentence &,vector<set<string> > &): void
Fries
adaboost
+classify(example,double[]): void
Omlet

Figure 2: FreeLing-2.1 Main Processing Classes.

- `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. Two modules are provided for this task. A fast and simple pattern–matching module based on capitalization (which yields an accuracy near 90%), and a NE recognizer based on the CoNLL-2002 shared task winning system (Carreras et al., 2002), rather slower, but with an accuracy over 92%.
- `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. FreeLing offers two PoS taggers with state-of-the-art accuracy (about 97%): One HMM–based following (Brants, 2000) and another based on relaxation labelling (Padró, 1998).
- NE `classifier`: Receives a list of `sentence` and classifies all `word` tagged as proper nouns in the given sentences. This module is based on the CoNLL-2002 shared task winning system (Carreras et al., 2002).
- `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 for each `word` selected `analysis`. This module is a direct inclusion of the UKB system (Agirre and Soroa, 2009).
- `chunk parser`: Receives a list of `sentence` and enriches each of them with a `parse_tree`. This module consists of a chart parser, and is a reimplementation of (Atserias and Rodríguez, 1998).
- `dependency parser`: Receives a list of parsed `sentence` and enriches each of them with a `dependency_tree`. This module uses a set of hand–written rules to build a dependency tree. Its original version is described in (Atserias et al., 2005), though several extensions to the expressive power of the rules have been added.
- `coreference solver`: Receives a document formed by parsed `sentence` and enriches it with coreference information. This module is based on the system proposed by (Soon et al., 2001).

5. Conclusions

We reviewed the first five years of FreeLing project, an open-source library of language analyzers, which is accomplishing its original goals of community participation, international dissemination, and utility both to academy and industry.

We also described the main enhancements included in version 2.1, and shallowly described the internal architecture of the library

Further work will involve consolidating the community, as well as keep including new functionalities and languages to the library.

Acknowledgements

This work has been partially funded by the Spanish Government via the KNOW2 (TIN2009-14715-C04-03/04) project.

6. 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 and Horacio Rodríguez. 1998. Tacat: Tagged corpus analizer tool. Technical report lsi-98-2-t, Departament de LSI. Universitat Politècnica de Catalunya.

Jordi Atserias, Josep Carmona, Irene Castellón, Sergi Cervell, Montserrat Civit, Lluís Màrquez, Mª Antònia Martí, Lluís Padró, Roberto Placer, Horacio Rodríguez, Mariona Taulé, and Jordi Turmo. 1998. Morphosyntactic analysis and parsing of unrestricted spanish text. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 1267–1274, Granada, Spain, May.

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.

Thorsten Brants. 2000. Tnt - a statistical part- of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing, ANLP*. ACL.

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

Xavier Carreras and Lluís Padró. 2002. A flexible distributed architecture for natural language analyzers. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC*, Las Palmas de Gran Canaria, Spain.

Xavier Carreras, Lluís Màrquez, and Lluís Padró. 2002. Named entity extraction using adaboost. In *Proceedings of CoNLL Shared Task*, pages 167–170, Taipei, Taiwan.

Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Montserrat Marimon, Núria Bel, Sergio Espeja, and Natalia Seghezzi. 2007. The spanish resource grammar: pre-processing strategy and lexical acquisition. In *Proceedings of the Workshop on Deep Linguistic Processing, Association for Computational Linguistics (ACL-DLP)*.

Lluís Padró, Samuel Reese, Eneko Agirre, and Aitor Soroa. 2010. Semantic services in freeling 2.1: Wordnet and ukb. In *Proceedings of the 5th International Conference of the Global WordNet Association (GWC-2010)*, Bombay, India, February.

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.