Language Processing Infrastructure in the XLike Project

Lluís Padró[*], Željko Agić[♠], Xavier Carreras[*], Blaz Fortuna[†], Esteban García-Cuesta[◇], Zhixing Li[⋆], Tadej Štajner[†], Marko Tadić[◁]

∗ TALP Research Center – Universitat Politècnica de Catalunya. Barcelona, Spain
♠ Linguistics Department, University of Postdam, Germany
† Jožef Stefan Institute. Ljubljana, Slovenia
◇ iSOCO S.A. – Edificio Tesla, Av Partenon 16-18. Madrid, Spain
⋆ Tsinghua University. Beijing, China
◁ Faculty of Humanities and Social Sciences – University of Zagreb. Zagreb, Croatia

Abstract

This paper presents the linguistic analysis tools and its infrastructure developed within the XLike project. The main goal of the implemented tools is to provide a set of functionalities for supporting some of the main objectives of XLike, such as enabling cross-lingual services for publishers, media monitoring or developing new business intelligence applications. The services cover seven major and minor languages: English, German, Spanish, Chinese, Catalan, Slovenian, and Croatian. These analyzers are provided as web services following a lightweight SOA architecture approach, and they are publically callable and are catalogued in META-SHARE.

Keywords: Language analysis tools, web services, text mining.

1. Introduction

The goal of the XLike project[1] is to develop technology which enables gathering documents in a variety of languages and genres (news, blogs, tweets, etc.) and extracting language-independent knowledge from them, in order to provide new and better services to publishers, media monitoring and business intelligence. In this line, the project use cases are provided by STA (Slovenian Press Agency) and Bloomberg. New York Times also participates as an associated partner.

Research partners in the project are Jožef Stefan Institute (JSI), Karlsruhe Institute of Technology (KIT), Universitat Politècnica de Catalunya (UPC), University of Zagreb (UZG), and Tsinghua University (THU). The Spanish company iSOCO is in charge of integration of all components developed in the project.

This paper deals with the language technology developed within the project XLike to convert input documents into a language-independent representation that facilitates later knowledge aggregation regardless of the source language in which the information was originally written.

To achieve this goal, a bench of linguistic processing pipelines needs to be devised as the first step in the document processing flow. Stages in these pipelines are homogeneous in all languages addressed in the project, and include tokenization, lemmatization, named entity detection and classification, dependency parsing, word sense disambiguation, and semantic role labeling.

All these analysis pipelines are deployed as web services, which enables multithreading, fast duplication, or redeployment if needed. The services are publicly callable and are described and cataloged in META-SHARE.[2] Note that *publicly callable* does not mean that the code is open-source (although most of the used components are) but that anyone can run a client program that submits documents for analysis.

2. Linguistic Analyzers

Apart from the basic state-of-the-art tokenizers, lemmatizers, PoS/MSD taggers, and NE recognizers, each pipeline requires deeper processors able to build the target language-independent semantic representation. For that, we rely on three steps: dependency parsing, semantic role labeling and word sense disambiguation. These three processes, combined with multilingual ontological resources such as different WordNets, are the key to the construction of our semantic representation.

2.1. Dependency Parsing

In XLike, we use the so-called graph-based methods for dependency parsing, introduced in (McDonald et al., 2005). In particular we use the following tools:

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

[2] http://www.meta-share.eu

Lang.	Treebank	Conversion to dependencies	#rels
en	Penn Treebank (Marcus et al., 1994)	CoNLL-09 (Hajič et al., 2009)	69
es	Ancora (Taulé et al., 2008)	Ancora	49
de	Tiger (Brants et al., 2004)	CoNLL-09	46
ca	Ancora (Taulé et al., 2008)	Ancora	50
sl	Učni (Holozan et al., 2008)	Učni	10
zh	CSDN (Mingqin et al., 2003)	CSDN	70
hr	HOBS (Tadić, 2007)	HOBS	70

Table 1: Corpora used to train dependency parsers.

- **Treeler**:[3] A library developed by the UPC team that implements several methods for dependency parsing, among other statistical methods for tagging and parsing. For dependency parsing, the implementation is based on (Carreras, 2007; Koo et al., 2008; Carreras et al., 2008), which in turn is based on the ideas by (McDonald et al., 2005).
- **MSTParser**:[4] This is the implementation provided by the authors of (McDonald et al., 2005; McDonald and Pereira, 2006). THU group uses this implementation to parse Chinese documents and UZG group to parse Croatian documents.

We use these tools in order to train dependency parsers for all XLike languages. Table 1 summarizes the treebanks used to develop the parsers. In some cases the treebanks are not in dependency format, so we use available tools to convert them. The third column indicates the method used to convert to dependencies, while the fourth column indicates the number of different kinds of dependency relations annotated in the corpus.

2.2. Semantic Role Labeling

As with syntactic parsing, we are using the Treeler library to develop machine-learning based SRL methods. In order to train models for this task, we use the treebanks made available by the CoNLL-2009 shared task (Hajič et al., 2009), which provided data annotated with predicate-argument relations for English, Spanish, Catalan, German and Chinese. No treebank annotated with semantic roles exists for Slovene or Croatian, thus, no SRL module is available for these languages in XLike pipelines.

For the languages where an SRL training corpus is available, a prototype of SRL has been integrated in the analysis pipeline.

The implemented method follows the architecture described in (Lluís et al., 2013).

[3] http://treeler.lsi.upc.edu

[4] http://sourceforge.net/projects/mstparser

2.3. Word Sense Disambiguation

The used Word Sense Disambiguation engine is the UKB (Agirre and Soroa, 2009) implementation provided by FreeLing (Padró and Stanilovsky, 2012). UKB is a non-supervised algorithm based on PageRank over a semantic graph such as WordNet.

Word sense disambiguation is performed for all languages for which a WordNet is publicly available. This includes all languages in the project except Chinese.

The goal of WSD is to map specific languages to a common semantic space, in this case, WN synsets. Thanks to existing connections between WN and other resources, SUMO and OpenCYC sense codes are also output when available.

Finally, we use PredicateMatrix (López de la Calle et al., 2014) — a lexical semantics resource combining WordNet, FrameNet, PropBank, and VerbNet — to project the obtained concepts to PropBank predicates and FrameNet diathesis structures, achieving a normalization of the semantic roles produced by the SRL (which are treebank-dependent, and thus, not the same for all languages).

2.4. Frame Extraction

The final step is to convert all the gathered linguistic information into a semantic representation. Our method is based on the notion of frame: a semantic frame is a schematic representation of a situation involving various participants. In a frame, each participant plays a role. There is a direct correspondence between roles in a frame and semantic roles; namely, frames correspond to predicates, and participants correspond to the arguments of the predicate. We distinguish three types of participants: entities, words, and frames.

For example, in the sentence in Figure 1, we can find three frames:

- *Base*: A person or organization being established or grounded somewhere. This frame has two participants: *Acme*, a participant of type entity play-

```
 1 Acme        acme        NP   B-PER  8 SBJ    _           _        A1      A0      A0
 2 ,           ,           Fc   O      1 P      _           _        _       _       _
 3 based       base        VBN  O      1 APPO   00636888-v  base.01  _       _       _
 4 in          in          IN   O      3 LOC    _           _        AM-LOC  _       _
 5 New_York    new_york    NP   B-LOC  4 PMOD   09119277-n  _        _       _       _
 6 ,           ,           Fc   O      1 P      _           _        _       _       _
 7 now         now         RB   O      8 TMP    09119277-n  _        _       AM-TMP  _
 8 plans       plan        VBZ  O      0 ROOT   00704690-v  plan.01  _       _       _
 9 to          to          TO   O      8 OPRD   _           _        _       A1      _
10 make        make        VB   O      9 IM     01617192-v  make.01  _       _       _
11 computer    computer    NN   O     10 OBJ    03082979-n  _        _       _       A1
12 and         and         CC   O     11 COORD  _           _        _       _       _
13 electronic  electronic  JJ   O     14 NMOD   02718497-a  _        _       _       _
14 products    product     NNS  O     12 CONJ   04007894-n  _        _       _       _
15 .           .           Fp   O      8 P      _           _        _       _       _
```

Figure 1: Output of the analyzers for the sentence *Acme, based in New York, now plans to make computer and electronic products.*

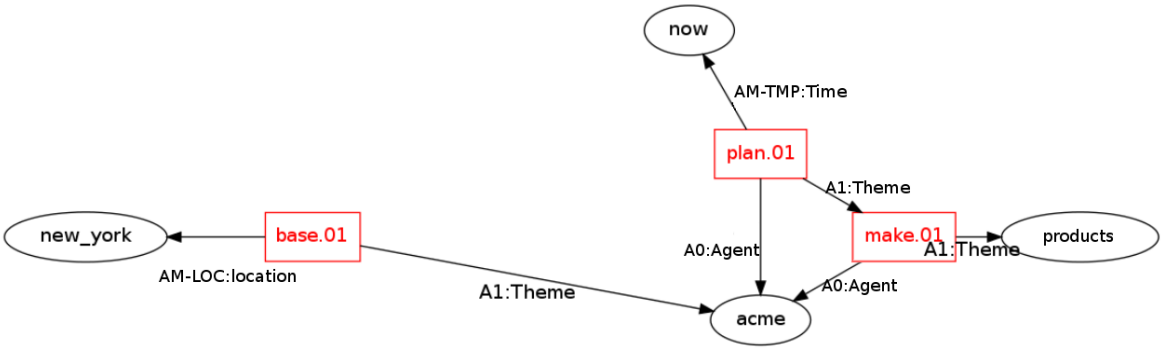

Figure 2: Graphical representation of frames in the example sentence.

ing the `theme` role (the thing being based), and *New York*, a participant of type entity playing the role of `location`.

- *Plan*: A person or organization planning some activity. This frame has three participants: *Acme*, a participant of type entity playing the `agent` role, *now*, a participant of type word playing the role of `time`, and *make*, a participant of type frame playing the `theme` role (i.e., the activity being planned).

- *Make*: A person or organization creating or producing something. Participants in this frame are: *Acme*, entity playing the `agent` role, and *products*, a participant of type word playing the `theme` role (i.e., the thing being created).

A graphical representation of the example sentence is presented in Figure 2.
It is important to note that frames are a more general representation than SVO-triples. While SVO-triples represent binary relations between two participants, frames can represent any n-ary relation. For example, the frame for *plan* is a ternary relation because it includes a temporal modifier. It is also important to note that frames can naturally represent higher-order relations: the theme of the frame *plan* is itself a frame, namely *make*.
Finally, although frames are extracted at sentence level, the resulting graphs are aggregated in a single semantic graph representing the whole document via a very simple co-reference resolution method based on detecting named entity aliases and repetitions of common nouns. Future improvements include using a state-of-the-art co-reference resolution module for languages where it is available.

3. Web Service Architecture Approach

The different language functionalities are represented in Figure 3 as different modules and are implemented following the service oriented architecture (SOA) approach defined in the project XLike.
Therefore all the pipelines (one for each language) have been implemented as web services and may be

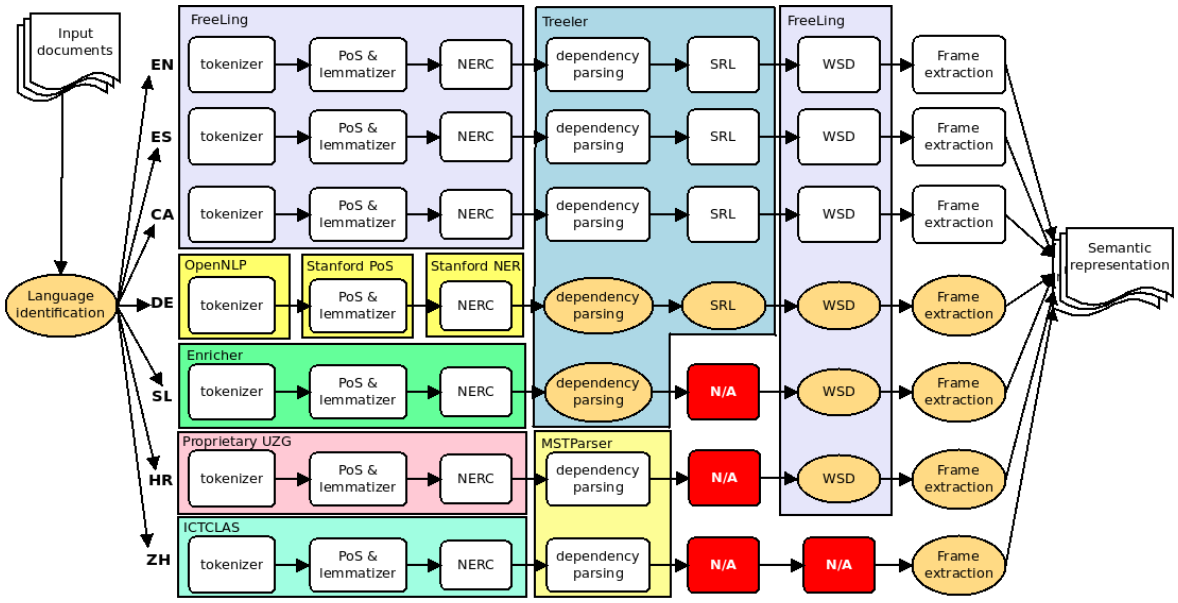

Figure 3: Xlike Language Processing Architecture.

requested to produce different levels of analysis (e.g. tokenization, lemmatization, NERC, parsing, relation extraction, etc.). This approach is very appealing due to the fact that it allows to treat every language independently and to execute the whole language analysis process at different threads or computers allowing an easier parallelization (e.g., using external high performance platforms such as Amazon Elastic Compute Cloud EC2[5]) as needed. Furthermore, it also provides independent development life-cycles for each language which is crucial in this type of research projects. Recall that these web services can be deployed locally or remotely, maintaining the option of using them in a stand-alone configuration.

Figure 3 also represents by large boxes the different technology used for the implementation of each module. White square modules indicates those functionalities that run locally inside a web service and can't be accessed directly, and shaded round modules indicate private web services which can be called remotely for accessing the specified functionality.

The main structure for each one of the pipelines is described below:

- **Spanish**, **English**, and **Catalan**: all modules are based on FreeLing (Padró and Stanilovsky, 2012) and Treeler.
- **German**: German shallow processing is based on OpenNLP,[6] Stanford POS tagger and NE extractor (Toutanova et al., 2003; Finkel et al., 2005). Dependency parsing, semantic role labeling, word sense disambiguation, and frame extraction are based on FreeLing and Treeler.
- **Slovene**: Slovene shallow processing is based on JSI Enrycher[7] (Štajner et al., 2010). The shallow processing pipeline in Enrycher consists of the Obeliks morphosyntactic analysis library (Grčar et al., 2012), the LemmaGen lemmatizer (Juršic et al., 2010) and a CRF-based entity extractor (Štajner et al., 2012). Dependency parsing and WSD are based on FreeLing and Treeler. Frame extraction is rule-based since no SRL corpus is available for Slovene.
- **Croatian**: Croatian shallow processing is on proprietary tokenizer, POS/MSD-tagging and lemmatization system (Agić et al., 2008), NERC system (Bekavac and Tadić, 2007) and dependency parser (Agić, 2012). WSD is based on FreeLing, and frame extraction is rule-based (no SRL corpus is available for Croatian).
- **Chinese**: Chinese shallow processing is based on ICTCLAS[8] word segmentation component. Deep processing consists of a semantic dependency parser trained on CSDN, and a WSD module based on TONGYICILIN (a Chinese synonym data-set containing 70,000 Chinese words). Frame extraction is rule-based.

Each language analysis service is able to process thousands of words per second when performing shal-

[5] http://aws.amazon.com/ec2/
[6] http://opennlp.apache.org
[7] http://enrycher.ijs.si
[8] http://ictclas.org/

low analysis (up to NE recognition), and hundreds of words per second when producing the semantic representation based on full analysis.

For instance, the average speed for analyzing an English document with shallow analysis (tokenizer, splitter, morphological analyzer, POS tagger, lemmatization, and NE detection and classification) is about 1,300 tokens/sec on a i7 3.4 Ghz processor (including communication overhead, XML parsing, etc.). This means that an average document (e.g, a news item of around 400 tokens) is analyzed in 0.3 seconds.

When using deep analysis (i.e., adding WSD, dependency parsing, and SRL to the previous steps), the speed drops to about 70 tokens/sec, thus an average document takes about 5.8 seconds to be analyzed.

The parsing and SRL models are still in a prototype stage, and we expect to largely reduce the difference between shallow and deep analysis times.

However, it is worth noting that the web-service architecture enables the same server to run a different thread for each client without using much extra memory. This exploitation of multiprocessor capabilities allows a parallelism degree of as many request streams as available cores, yielding an actually much higher average speed when large collections must be processed.

4. Applications

The presented linguistic analysis infrastructure has been used in the news press and social media domains. For this purpose, the Newsfeed tool (Mitja and Novak, 2012) has collected a clean, continuous, and real time aggregated stream of mainstream news articles and blog posts from RSS-enabled websites. Each of the articles is processed in real-time by the presented architecture obtaining a multilingual linguistically annotated set of articles which later on are consumed for providing cross searching capabilities to various applications.[9]

An example of such application, which was also evaluated, is assisting news editors in in discovering articles relevant to their area, which are published in different languages. For example, Slovenian Press Agency monitoring foreign press related to Slovenia. We found significant improvements on time and on coverage due to the easy accessibility to a larger number of articles. Another application build on top of presented infrastructure is Event Registry, which extracts events from news articles.[10]

[9] http://sandbox-xlike.isoco.com/portal

[10] http://eventregistry.org/

5. Conclusion

We presented the web-service based architecture used in XLike FP7 project to linguistically analyze large amounts of documents in seven different languages. The analysis pipelines perform basic processing as tokenization, PoS-tagging, and named entity extraction, as well as deeper analysis such as dependency parsing, word sense disambiguation, and semantic role labeling. The result of these linguistic analyzers is a semantic graph capturing the main events described in the document and their core participants.

This semantic representation is later used in XLike for document mining use cases such as enabling cross-lingual services for publishers, media monitoring or developing new business intelligence applications.

The developed web services are publicly callable and are described in META-SHARE.

6. References

Agić, Ž., Tadić, M., and Dovedan, Z. (2008). Improving part-of-speech tagging accuracy for Croatian by morphological analysis. *Informatica*, 32(4):445–451.

Agić, Ž. (2012). K-best spanning tree dependency parsing with verb valency lexicon reranking. In *Proceedings of COLING 2012: Posters*, pages 1–12, Mumbai, India, December. The COLING 2012 Organizing Committee.

Agirre, E. and Soroa, A. (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.

Bekavac, B. and Tadić, M. (2007). Implementation of Croatian NERC system. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing (BSNLP2007), Special Theme: Information Extraction and Enabling Technologies*, pages 11–18. Association for Computational Linguistics.

Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. (2004). Tiger: Linguistic interpretation of a german corpus. *Journal of Language and Computation*, (2):597–620.

Carreras, X., Collins, M., and Koo, T. (2008). Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 9–16, Manchester, England, August. Coling 2008 Organizing Committee.

Carreras, X. (2007). Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 957–961, Prague, Czech Republic, June.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, pages 363–370.

Grčar, M., Krek, S., and Dobrovoljc, K. (2012). Obeliks: statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik. In *Zbornik Osme konference Jezikovne tehnologije*, Ljubljana, Slovenia.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., et al. (2009). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009): Shared Task*, pages 1–18, Boulder, Colorado, USA, June.

Holozan, P., Krek, S., Pivec, M., Rigač, S., Rozman, S., and Velušček, A. (2008). Specifikacije za učni korpus. Technical report, Projekt 'Sporazumevanje v slovenskem jeziku' ESS in MŠŠ.

Juršic, M., Mozetic, I., Erjavec, T., and Lavrac, N. (2010). Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio. Association for Computational Linguistics.

Lluís, X., Carreras, X., and Màrquez, L. (2013). Joint arc-factored parsing of syntactic and semantic dependencies. *Transactions of the Association for Computational Linguistics*, 1:219–230.

López de la Calle, M., Laparra, E., and Rigau, G. (2014). First steps towards a predicate matrix. In *Proceedings of the Global WordNet Conference (GWC 2014)*, Tartu, Estonia, January. GWA.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19.

McDonald, R. and Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 81–88.

McDonald, R., Crammer, K., and Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, Ann Arbor, Michigan.

Mingqin, L., Juanzi, L., Zhendong, D., Zuoying, W., and Dajin, L. (2003). Building a large chinese corpus annotated with semantic dependency. In *Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17*, SIGHAN '03, pages 84–91, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mitja, T. and Novak, B. (2012). Internals of an aggregated web news feed. In *Proceedings of the 15th International Multiconference on Information Society IS-2012*.

Padró, L. and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

Štajner, T., Rusu, D., Dali, L., Fortuna, B., Mladenić, D., and Grobelnik, M. (2010). A service oriented framework for natural language text enrichment. *Informatica*, 34(3):307–313.

Tadić, M. (2007). Building the Croatian Dependency Treebank: the initial stages. *Suvremena lingvistika*, 33(63):85–92.

Taulé, M., Martí, M. A., and Recasens, M. (2008). Ancora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of 6th International Conference on Language Resources and Evaluation*, Marrakesh, Morocco, May.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Lin- guistics on Human Language Technology (NAACL'03)*.

Štajner, T., Erjavec, T., and Krek, S. (2012). Razpoznavanje imenskih entitet v slovenskem besedilu. In *In Proceedings of 15th Internation Multiconference on Information Society - Jezikovne Tehnologije*, Ljubljana, Slovenia.