Multilingual, Efficient and Easy NLP Processing with IXA Pipeline

Rodrigo Agerri
IXA NLP Group
Univ. of the Basque Country
UPV/EHU
Donostia San-Sebastián
rodrigo.agerri@ehu.es

Josu Bermudez
Deusto Institute of Technology
Deustotech
Univ. of Deusto
Bilbao
josu.bermudez@deusto.es

German Rigau
IXA NLP Group
Univ. of the Basque Country
UPV/EHU
Donostia-San Sebastián
german.rigau@ehu.es

Abstract

IXA pipeline is a modular set of Natural Language Processing tools (or pipes) which provide easy access to NLP technology. It aims at lowering the barriers of using NLP technology both for research purposes and for small industrial developers and SMEs by offering robust and efficient linguistic annotation to both researchers and non-NLP experts. IXA pipeline can be used "as is" or exploit its modularity to pick and change different components. This paper describes the general data-centric architecture of IXA pipeline and presents competitive results in several NLP annotations for English and Spanish.

1 Introduction

Many Natural Language Processing (NLP) applications demand some basic linguistic processing (Tokenization, Part of Speech (POS) tagging, Named Entity Recognition and Classification (NER), Syntactic Parsing, Coreference Resolution, etc.) to be able to further undertake more complex tasks. Generally, NLP annotation is required to be as accurate and efficient as possible and existing tools, quite righly, have mostly focused on performance. However, this generally means that NLP suites and tools usually require researchers to do complex compilation/installation/configuration in order to use such tools. At the same time, in the industry, there are currently many Small and Medium Enterprises (SMEs) offering services that one way or another depend on NLP annotations.

In both cases, in research and industry, acquiring, deploying or developing such base qualifying technologies is an expensive undertaking that redirects their original central focus: In research, much time is spent in the preliminaries of a particular research experiment trying to obtain the required basic linguistic annotation, whereas in an industrial environment SMEs see their already limited resources taken away from offering products and services that the market demands. IXA pipeline provides *ready to use modules* to perform efficient and accurate linguistic annotation to allow users to focus on their original, central task. When designing the architecture, we took several decisions with respect to what IXA pipeline had to be:

Simple and ready to use: Every module of the IXA pipeline can be up an running after two simple steps.

Portable: The modules come with "all batteries included" which means that no classpath configurations or installing of any third-party dependencies is required. The modules will will run on any platform as long as a JVM 1.7+ and/or Python 2.7 are available.

Modular: Unlike other NLP toolkits, which often are built in a monolithic architecture, IXA pipeline is built in a data centric architecture so that modules can be picked and changed (even from other NLP toolkits). The modules behave like Unix pipes, they all take standard input, do some annotation, and produce standard output which in turn is the input for the next module. The data-centric architecture of IXA pipeline means that any module is *highly independent* and can therefore be used with other tools from other toolkits if required.

Efficient: Piping the tokenizer (250K words per second) POS tagger and lemmatizer all in one process annotates over 5K words/second. The NERC module annotates over 5K words/second. In a multi-core machine, these times are dramatically reduced due to multi-threading.

Multilingual: Currently we offer NLP annotations for both English and Spanish, but other languages are being included in the pipeline. Tokenization already works for several languages, including Dutch, French, Italian, German, Spanish and English.

Accurate: For example, POS tagging and NERC for English and Spanish are comparable with other state of the art systems, as it is the coreference resolution module for English.

Apache License 2.0: IXA Pipeline is licensed under the Apache License 2.0, an open-source license that facilitates source code use, distribution and integration, also for commercial purposes.[1]

Next section describes the *IXA pipeline* architecture, section 3 the modules so far developed. Whenever available, we also present empirical evaluation. Section 4 describes the various ways of using the tools. Finally, section 5 discusses some concluding remarks.

[1] http://www.apache.org/licenses/LICENSE-2.0.html

2 Architecture

IXA pipeline is primarily conceived as a set of *ready to use tools* that can provide efficient and accurate linguistic annotation without any installation/configuration/compilation effort. As in Unix-like operative systems, IXA pipeline consists of a set of processes chained by their standard streams, in a way that the output of each process feeds directly as input to the next one. The Unix pipeline metaphor has been applied for NLP tools by adopting a very simple and well known data centric architecture, in which every module/pipe is interchangeable for another one as long as it takes and produces the required data format.

The data format in which both the input and output of the modules needs to be formatted to represent and filter linguistic annotations is KAF (Bosma et al., 2009). KAF is a language neutral annotation format representing both morpho-syntactic and semantic annotation in a structured format. KAF was originally designed in the Kyoto European project[2], but it has since been in continuous development[3]. Our Java modules all use kaflib[4] library for easy integration.

Every module in the IXA pipeline, except the coreference resolution, is implemented in Java, and requires Java JDK1.7+ to compile. The integration of the Java modules in the IXA pipeline is performed using Maven[5]. Maven is used to take care of classpaths configurations and third-party tool dependencies. This means that the binaries produced and distributed *will work off-the-self*. The coreference module uses pip[6] to provide an easy, one step installation. If the source code of an ixa-pipe-$module is cloned from the remote repository, *one command* to compile and have ready the tools will suffice.

Some modules in IXA pipeline provide linguistic annotation based on probabilistic supervised approaches such as POS tagging, NER and Syntactic Parsing. IXA pipeline uses two well known machine learning algorithms, namely, Maximum Entropy and the Perceptron. Both Perceptron (Collins, 2002; Collins, 2003) and Maximum Entropy models (Ratnaparkhi, 1999) are adaptable algorithms which have been successfully applied to NLP tasks such as POS tagging, NER and Parsing with state of the art results. To avoid duplication of efforts, IXA pipeline uses the already available open-source Apache OpenNLP API[7] to train POS, NER and parsing probabilistic models using these two approaches.

[2]http://kyoto-project.eu
[3]http://www.opener-project.org/kaf/
[4]https://github.com/ixa-ehu/kaflib
[5]http://maven.apache.org/
[6]https://pypi.python.org/pypi/pip
[7]http://opennlp.apache.org

3 Pipes

IXA pipeline currently provides the following linguistic annotations: Sentence segmentation, tokenization, Part of Speech (POS) tagging, Lemmatization, Named Entity Recognition and Classification (NER), Constituent Parsing and Coreference Resolution. Every module works for English and Spanish and is implemented in Java/Maven as described above. The only exception is the coreference resolution module, which currently is available in Python 2.7 and for English only (Spanish version will comme soon). We will now describe which annotation services are provided by each module of the pipeline.

3.1 ixa-pipe-tok

This module provides rule-based Sentence Segmentation and Tokenization for French, Dutch, English, Italian and Spanish. It produces tokenized and segmented text in KAF, running text and CoNLL formats. The rules are originally based on the Stanford English Tokenizer[8], but with substantial modifications and additions. These include tokenization for other languages such as French and Italian, normalization according the Spanish Ancora Corpus (Taulé et al., 2008), paragraph treatment, and more comprehensive gazeteers of non breaking prefixes. The tokenizer depends on a JFlex[9] specification file which compiles in seconds and performs at a very reasonable speed (around 250K word/second, and much quicker with Java multithreading).

3.2 ixa-pipe-pos

ixa-pipe-pos provides POS tagging and lemmatization for English and Spanish. We have obtained the best results so far with the same featureset as in Collins's (2002) paper. *Perceptron* models for English have been trained and evaluated on the WSJ treebank using the usual partitions (e.g., as explained in Toutanova et al. (2003). We currently obtain a performance of 97.07% vs 97.24% obtained by Toutanova et al., (2003)). For Spanish, *Maximum Entropy* models have been trained and evaluated using the Ancora corpus; it was randomly divided in 90% for training and 10% for testing. This corresponds to 440K words used for training and 70K words for testing. We obtain a performance of 98.88% (the corpus partitions are available for reproducibility). Giménez and Marquez (2004) report 98.86%, although they train and test on a different subset of the Ancora corpus.

Lemmatization is currently performed via 3 different dictionary lookup methods: (i) *Simple Lemmatizer*: It is based on HashMap lookups on a plain text dictionary. Currently we use dictionaries from the LanguageTool project[10] under their distribution licenses. The English

[8]http://www-nlp.stanford.edu/software/tokenizer.shtml
[9]http://jflex.de/
[10]http://languagetool.org/

dictionary contains 300K lemmas whereas the Spanish provides over 600K; (ii) *Morfologik-stemming*[11]: The Morfologik library provides routines to produce binary dictionaries, from dictionaries such as the one used by the Simple Lemmatizer above, as finite state automata. This method is convenient whenever lookups on very large dictionaries are required because it reduces the memory footprint to 10% of the memory required for the equivalent plain text dictionary; and (iii) We also provide lemmatization by lookup in *WordNet-3.0* (Fellbaum and Miller, 1998) via the JWNL API[12]. Note that this method is only available for English.

3.3 ixa-pipe-nerc

Most of the NER systems nowdays consist of language independent systems (sometimes enriched with gazeteers) based on automatic learning of statistical models. *ixa-pipe-nerc* provides Named Entity Recognition (NER) for English and Spanish. The named entity types are based on the CONLL 2002[13] and 2003[14] tasks which were focused on language-independent supervised named entity recognition (NER) for four types of named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. We currently provide two very fast language independent models using a rather simple baseline featureset (e.g., similar to that of Curran and Clark (2003), except POS tag features).

For English, perceptron models have been trained using CoNLL 2003 dataset. We currenly obtain 84.80 F1 which is coherent with other results reported with these features (Clark and Curran, 2003; Ratinov and Roth, 2009). The best Stanford NER model reported on this dataset achieves 86.86 F1 (Finkel et al., 2005), whereas the best system on this dataset achieves 90.80 F1 (Ratinov and Roth, 2009), using non local features and substantial external knowledge.

For Spanish we currently obtain best results training Maximum Entropy models on the CoNLL 2002 dataset. Our best model obtains 79.92 F1 vs 81.39 F1 (Carreras et al., 2002), the best result so far on this dataset. Their result uses external knowledge and without it, their system obtains 79.28 F1.

3.4 ixa-pipe-parse

ixa-pipe-parse provides statistical constituent parsing for English and Spanish. Maximum Entropy models are trained to build shift reduce bottom up parsers (Ratnaparkhi, 1999) as provided by the Apache OpenNLP API. Parsing models for English have been trained using the Penn treebank and for Spanish using the Ancora corpus (Taulé et al., 2008).

Furthermore, *ixa-pipe-parse* provides two methods of HeadWord finders: one based on Collins' head rules as defined in his PhD thesis (1999), and another one based on Stanford's parser Semantic Head Rules[15]. The latter are a modification of Collins' head rules according to lexical and semantic criteria. These head rules are particularly useful for the Coreference resolution module and for projecting the constituents into dependency graphs.

As far as we know, and although previous approaches exist (Cowan and Collins, 2005), *ixa-pipe-parse* provides the first publicly available statistical parser for Spanish.

3.5 Coreference Resolution

The module of coreference resolution included in the IXA pipeline is loosely based on the Stanford Multi Sieve Pass system (Lee et al., 2013). The module takes every linguistic information it requires from the KAF layers annotated by all the previously described modules. The system consists of a number of rule-based sieves. Each sieve pass is applied in a deterministic manner, reusing the information generated by the previous sieve and the mention processing. The order in which the sieves are applied favours a highest precision approach and aims at improving the recall with the subsequent application of each of the sieve passes. This is illustrated by the evaluation results of the CoNLL 2011 Coreference Evaluation task (Lee et al., 2013), in which the Stanford's system obtained the best results.

So far we have evaluated our module on the CoNLL 2011 testset and we are a 5% behind the Stanford's system (52.8 vs 57.6 CoNLL F1), the best on that task (Lee et al., 2013). It is interesting that in our current implementation, mention-based metrics are favoured (CEAF and B^3). Still, note that these results are comparable with the results obtained by the best CoNLL 2011 participants. Currently the module performs coreference resolution only for English, although a Spanish version will be coming soon.

4 Related Work

Other NLP toolkits exist providing similar or more extensive functionalities than the IXA pipeline tools, although not many of them provide multilingual support. GATE (Cunningham, 2002) is an extensive framework supporting annotation of text. GATE has some capacity for wrapping Apache UIMA components[16], so should be able to manage distributed NLP components. However, GATE is a very large and complex system, with a corresponding steep learning curve.

Freeling (Padró and Stanilovsky, 2012) provides multilingual processing for a number of languages, incluing Spanish and English. As opposed to IXA pipeline, Freeling is a monolithic toolkit written in C++ which needs to be compiled natively. The Stanford

[11] https://github.com/morfologik/morfologik-stemming
[12] http://jwordnet.sourceforge.net/
[13] http://www.clips.ua.ac.be/conll2002/ner/
[14] http://www.clips.ua.ac.be/conll2003/ner/
[15] http://www-nlp.stanford.edu/software/lex-parser.shtml
[16] http://uima.apache.org/

CoreNLP[17] is a monolithic suite, which makes it difficult to integrate other tools in its chain.

IXA pipeline tools can easily be used piping the input with the output of another too, and it is also possible to easily replace or extend the toolchain with a third-party tool. IXA pipeline is already being used to do extensive parallel processing in the FP7 European projects OpeNER[18] and NewsReader[19].

5 Conclusion and Future Work

IXA pipeline provides a simple, efficient, accurate and ready to use set of NLP tools. Its modularity and data centric architecture makes it flexible to pick and change or integrate new linguistic annotators. Currently we offer linguistic annotation for English and Spanish, but more languages are being integrated. Furthermore, other annotations such as Semantic Role Labelling and Named Entity Disambiguation are being included in the pipeline following the same principles.

Additionally, current integrated modules are being improved: both on the quality and variety of the probabilistic models, and on specific issues such as lemmatization, and treatment of time expressions. Finally, we are adding server-mode execution into the pipeline to provide faster processing. IXA pipeline is publicly available under Apache 2.0 license: *http://adimen.si.ehu.es/web/ixa-pipes*.

Acknowledgements

TThis work has been supported by the OpeNER FP7 project under Grant No. 296451, the FP7 NewsReader project, Grant No. 316404, and by the SKATER Spanish MICINN project No TIN2012-38584-C06-01. The work of Josu Bermudez on coreference resolution is supported by a PhD Grant of the University of Deusto (http://www.deusto.es).

References

Wauter Bosma, Piek Vossen, Aitor Soroa, German Rigau, Maurizio Tesconi, Andrea Marchetti, Monica Monachini, and Carlo Aliprandi. 2009. Kaf: a generic semantic annotation format. In *Proceedings of the GL2009 Workshop on Semantic Annotation*.

X. Carreras, L. Marquez, and L. Padro. 2002. Named entity extraction using AdaBoost. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4.

Stephen Clark and James Curran. 2003. Language Independent NER using a Maximum Entropy Tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada.

[17] http://nlp.stanford.edu/software/corenlp.shtml

[18] http://www.opener-project.org

[19] http://www.newsreader-project.eu

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

Brooke Cowan and Michael Collins. 2005. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 795–802. Association for Computational Linguistics.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

C. Fellbaum and G. Miller, editors. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press, Cambridge (MA).

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Jesús Giménez and Lluis Marquez. 2004. Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54, January.

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

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, page 147155.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine learning*, 34(1-3):151–175.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for catalan and spanish. In *LREC*.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pages 252–259.