A Modular Chain of NLP Tools for Basque

Arantxa Otegi, Nerea Ezeiza, Iakes Goenaga, and Gorka Labaka

IXA Group, University of the Basque Country, UPV/EHU
arantza.otegi@ehu.eus

Abstract. This work describes the initial stage of designing and implementing a modular chain of Natural Language Processing tools for Basque. The main characteristic of this chain is the deep morphosyntactic analysis carried out by the first tool of the chain and the use of these morphologically rich annotations by the following linguistic processing tools of the chain. It is designed following a modular approach, showing high ease of use of its processors. Two tools have been adapted and integrated to the chain so far, and are ready to use and freely available, namely the morphosyntactic analyzer and PoS tagger, and the dependency parser. We have evaluated these tools and obtained competitive results. Furthermore, we have tested the robustness of the tools on an extensive processing of Basque documents in various research projects.

1 Introduction

We live immersed in the Information Society and we have access to a vast amount of information mostly in the form of text. It is increasingly necessary to incorporate the automatic processing of information and consequently, of languages, since the medium in which this information is mostly found is in natural language. Although English is the predominant language in the current globalized environment, information is multilingual, and the presence of minor languages, like Basque, is increasing.

Basque presents a complex intraword morphological structure based on morpheme agglutination. Regarding the agglutinative nature of Basque, it could be said that the affixes are attached to the lemmas, and each attached morpheme carries (ordinarily) only one meaning. For instance, in *mendian* (Basque for 'in the mountain'), *mendi* stands for 'mountain', *–a* for the determiner (translatable as 'the'), and *–n* for the locative case. The determiner, number and declension case morphemes are appended to the last element of the noun phrase and always occur in this order. In addition, often the syntactic function of a word is conveyed by the suffix attached to it as a case marker. In the previous example, the locative case added to the stem *mendi* ('mountain') assigns the verb complement (locative) function to the word. However, not all the lexical words in a phrase are inflected in Basque.

Moreover, word formation is very productive, as it is very usual to create new compounds as well as derivatives. As a result of the wealth of information contained within word forms, complex structures have to be built to represent

complete morphological information at word level. For example, the compound *oxigeno-hornitzailea* can be segmented into: a) the sequence *oxigeno* (Basque for 'oxygen'), the hyphen, *hornitzaile* ('provider') and *–a* (meaning 'the person who provides oxygen'); b) *oxigeno*, the hyphen, *horni* (stem of the verb *hornitu* 'provide'), the lexical suffix *–tzaile* (attached to a verb 'a person or thing that'), and *–a*. This last segmentation is ambiguous in that the derivation affix might be modifying only the verb stem, meaning the same as the previous segmentation, or modifying the compound meaning 'the device that provides oxygen' or 'oxygen bottle'.

The rules to combine the intraword information to represent the final morphosyntactic interpretation are defined via a word grammar [1].

Having all these characteristics in mind, the main characteristic of the chain is the ease to integrate complex processors, such as the deep morphosyntactic analysis carried out by the first tool. Another important feature is the use of morphologically rich annotations to transmit information through the chain. Finally, it is modular to let users decide which processors to include in the chain.

In this paper we introduce the already developed first two processors of the chain: the morphological analyzer and a PoS tagger (ixa-pipe-pos-eu), and the dependency parser (ixa-pipe-dep-eu). In the future, the chain will be extended in order to make it possible to carry out the whole linguistic processing for Basque, as it will provide not only basic processing tools like tokenization, but also more complex tasks as lemmatization, part-of-speech (PoS) tagging, syntactic parsing, already included, as well as Named Entity Recognition and Classification, Named Entity Disambiguation, coreference resolution, semantic role labeling, etc.

The rest of the paper is organized as follows. Next section presents some previous similar work. Section 3 introduces the general description of the modular processing chain for Basque. Section 4 describes the tools so far developed, and we also present their empirical evaluation. Section 5 presents some projects in which these tools have been used. Finally, section 6 discusses some concluding remarks and future work.

2 Related Work

Many NLP toolkits exist providing extensive functionalities, like GATE [7], Stanford CoreNLP [12] and Freeling [13]. They are large and complex system, making difficult the use or the integration of other tools in their chain. Conversely, IXA pipes is intended to be simple, ready to use, modular and portable, as well as efficient, multilingual, accurate and open source [4].

The most similar work to our approach is IXA pipes, as it is a modular set of multilingual NLP tools which is publicly available.[1] It offers robust and efficient linguistic annotation to both researchers and non-NLP experts with the aim of lowering the barriers of using NLP technology either for research purposes or

[1] `http://ixa2.si.ehu.eus/ixa-pipes/`

for small industrial developers. It provides several linguistic annotation tools, including, among others, a tokenizer and a module for PoS tagging and lemmatizing, which are the tools we focus on in this work. Additionally, it offers third party tools for other linguistic annotations. Currently, it supports a different number of languages for each tool. For instance, it offers tokenization, PoS tagging and lemmatizing for Basque, Dutch, English, French, Galician, German, Italian and Spanish.

The morphosyntactic analysis and PoS tagger tool presented in this work is based on Eustagger, a robust lemmatizer/tagger for Basque [8], which integrates the word grammar described in [1]. This tagger has been extensively used during the last 20 years to process Basque corpora. Among them, we want to highlight EPEC [2], an annotated corpus for Basque, aimed to be a reference corpus for the development and improvement of several NLP tools for Basque. EPEC was initially a 50,000-word sample collection of written standard Basque that has afterwards been extended to 300,000 words. This corpus has been automatically processed and the results have been manually revised. Only half of the complete collection has been annotated with syntactic information yet and part of it (80% according to [6]) has been automatically converted to Universal Dependencies.[2]

It is worth mentioning that the lemmatizer for Basque in IXA pipes (ixa-pipe-pos) uses a model trained on the mentioned Basque Universal dependency corpus, as opposed to the morphological processor integrated in the chain presented in this work (see Section 4.1). Being both approaches different, we intend to compare their performance under the same conditions to see the strengths and weaknesses of each one.

Apart from being modular, our approach has another characteristic in common with IXA pipes: the input/output format of the tools. As all the modules in both set of tools read and write NAF format (see Section 3), it is possible the interaction between them. For example, it is possible to extend the modular chain of Basque processing tools with a tool for NERC in IXA pipes (ixa-pipe-nerc), which is ready for Basque (among other several languages) [5].

3 General Description of the Modular Chain

The linguistic processing tools for Basque integrated in the modular chain, and described in Section 4, were initially designed for internal use and each of the tools was designed to run independently from the rest. Thus, the tools have been adapted to follow the main characteristics of the modular chain described below.

One of the main features of the chain is its modularity. That is, the tools can be picked and changed, as long as they read and write the required data format via the standard streams. The processors interact like Unix pipes, specifically they all take standard input, do some linguistic processing, and produce standard output which feeds directly the next one.

The data format used to represent and pipe linguistic annotations in such modular chain is NAF [9], a linguistic annotation format designed for complex

[2] `http://universaldependencies.org/#eu`

```
cat input.txt | sh ixa-pipe-pos-eu/run.sh | sh ixa-pipe-dep-eu/run.sh
```

Fig. 1. Command line invocation for running the two tools of the modular chain.

NLP pipelines.[3] In that way, by default, the input and output of all the tools is formatted in NAF, except for the input of the first one, ixa-pipe-pos-eu, which takes raw text as input. The annotation chain can be applied to any text, such as a single sentence, a paragraph or whole story. All the tools work with UTF-8 character encoding.

Another main feature of the chain is the minimal compilation or installation effort in order to get started using the tools. Besides, running the processing chain is simple and is done by a command-line interface. Once you get the tools, without doing any installation or configuration, doing linguistic processing for a file can be as easy as Figure 1 shows. Using the command displayed in the example, a raw text file is processed, first, analyzing morphologically and PoS tagging, and next, applying a dependency parser. The linguistic annotations provided by the whole chain will be written through standard output, formatted in NAF. In addition, some tools could have their own properties to allow further customization of their usage.

Additionally, the tools (binary tarballs) are publicly available[4] and are distributed under a free software license, GPL v3.[5]

4 The Tools Integrated in the Modular Chain

4.1 ixa-pipe-pos-eu

ixa-pipe-pos-eu is a robust and wide-coverage morphological analyzer and a PoS tagger, which is an adapted version of Eustagger, a tool lemmatizer/tagger for Basque [8]. It is the first module of the linguistic processing chain. The tool takes a raw text as an input text and outputs the lemma, the PoS tag and the morphological information for each token in NAF format.

It processes a text morphosyntactically following the next steps: tokenization, segmentation, the word grammar, treatment of multiword expressions and morphosyntactic disambiguation.

The morphological segmentation of words is based on a set of two-level rules converted into finite-state transducers. The analysis is performed in two main phases and gives as a result all the possible analyses of each word in the text: on the one hand, the standard analyzer that is able to analyze/generate standard-language words based on a general lexicon and the corresponding rules for morphotactics and morphophonological changes; on the other hand, the guesser or analyzer of words with lemmas not belonging to the previous lexicon. For the

[3] http://wordpress.let.vupr.nl/naf/

[4] http://ixa2.si.ehu.es/ixakat/

[5] http://www.gnu.org/licenses/gpl-3.0.en.html

guesser the lexicon is simplified by allowing only open categories (nouns, adjectives, verbs, etc.) and any combination of characters as lemmas. This general set of lemmas is combined with affixes related to open categories and general rules in order to capture as many morphologically significant features as possible. In this two-phase architecture, each word will be processed by the first analyzer that is able to produce at least one valid segmentation following the order defined before. Comparing to Eustagger, this adaptation lacks of the module for linguistic variants, which has been discarded to reduce the complexity of the process and to make it more efficient.

After segmenting the word into its constituent morphemes, the word grammar based processor analyzes and elaborates the sequential intraword information in order to build the information of the word as a whole.

For the detection of multiword expression, we have integrated a reduced version of the processor in Eustagger due to simplicity, which detects only the most common expressions.

The performance of the morphosyntactic analysis in Eustagger is very good, assigning the correct analysis to 99% of the words despite the high ambiguity (each token has 3.56 reading on average, and 1.56 reading taking only PoS tag into account). Although we still have to confirm the results of this reduced tool in question, we expect to obtain similar results.

Once we have given all possible morphological analysis to each token/multitoken, PoS tagging and lemmatization must be performed in order to assign the correct lemma and grammatical category to each token taking into account the context. The disambiguation is based on linguistic knowledge, as well as statistical information. First, a set of Constraint Grammar rules [11] are used to discard some analysis. After that, a stochastic HMM disambiguation is applied to choose the final analysis. The HMM model has been trained using the training set of the EPEC corpus (100,000 words), and the CG rules have also been defined and tuned based on the same set of the corpus.

Eustagger has been evaluated on the test set of EPEC corpus (50,000 words) obtaining a performance of 95.17% on PoS tagging accuracy, and 91.89% when considering all morphological information. As we have noted before, we expect to obtain similar figures for ixa-pipe-pos-eu, because we consider that the differences in the processing (lack of variants and smaller set of multiword expressions) should not distort too much the results of this robust approach.

4.2 ixa-pipe-dep-eu

ixa-pipe-dep-eu is a dependency parser which takes a NAF document containing lemmas, PoS tags and morphological annotations from the output of the previous ixa-pipe-pos-eu tool.

There are two main approaches for dependency parsing: transition-based and graph-based. And these are the state of the art dependency parsers: Malt-

Parser[6], MaltOptimizer[7], MST Parser[8] and Mate.[9] The two former ones follow a transition-based approach, whereas the two latter ones follow a graph-based approach.

Following a transition-based approach, MaltParser and MaltOptimizer (MaltParser optimization tool) consist of a transition system for deriving dependency trees, coupled with a classifier for deterministically predicting the next transition given a feature representation of the current parser configuration. The main difference between them is that MaltOptimizer first performs an analysis of the training set in order to select a suitable starting point for optimization, and then guides the user through the optimization of parsing algorithm, feature model, and learning algorithm.

In contrast, and following a graph-based approach, MST Parser adopts the second order maximum spanning tree dependency parsing algorithm. A maximum spanning tree dependency based parser decomposes a dependency structure into parts known as factors. The factors of the first order maximum spanning tree parsing algorithm are edges consisting of the head, the dependent (child) and the edge label. The second order parsing algorithm uses the edges to those children which are closest to the dependent, the child of the dependent occurring in the sentence between the head and the dependent, and the edge to a grandchild, in addition to the first order factors.

ixa-pipe-dep-eu tool is based on the graph-based version of Mate parser, which adopts the second order maximum spanning tree dependency parsing algorithm, as does MST Parser. The main two difference between them are the following: 1) Mate parser considers more children and edges in order to create dependency trees and 2) it uses a new parallel parsing and feature extraction algorithm that improves accuracy as well as parsing speed. Our dependency parsing tool has been trained using part of the Basque Dependency Treebank [3] which contains 96,000 tokens and 7,700 sentences.

We have evaluated ixa-pipe-dep-eu using part of the Basque Dependency Treebank (13,851 tokens) and it have obtained 83.00% LAS (Labeled Attachment Score) [10], which is a good result taking into account that Basque is a morphologically rich language. Having said that, we consider interesting to compare ixa-pipe-dep-eu with the other state of the art statistical parsers mentioned above. The results of all the systems are shown in Table 1. It can be observed that our dependency parsing tool outperforms all the rest of the parsers, yet there is not significant difference between ixa-pipe-dep-eu and MST Parser.

Although the results of ixa-pipe-dep-eu are promising, there is room for improvement. For this reason, on the one hand, we are trying to increase the size of our training set, and on the other hand, we are studying the use of feature engineering in order to select only the most beneficial morphological features for dependency parsing of Basque.

---

[6] `http://www.maltparser.org/`

[7] `http://nil.fdi.ucm.es/maltoptimizer/install.html`

[8] `http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html`

[9] `https://code.google.com/archive/p/mate-tools/`

Table 1. Results of different dependency parsers for Basque.

Dependency parser	LAS
MaltParser	77.78%
MaltParser + MaltOptimizer	80.04%
MST Parser	82.69%
ixa-pipe-dep-eu	83.00%

5 Some Projects Using the Modular Chain

The modular chain of NLP tools for Basque is already being used successfully for the linguistic annotations in several projects.

One of these projects is QTLeap,[10] a FP7 European project on machine translation. QTLeap project explores novel ways for attaining machine translation of higher quality that are opened by a new generation of increasingly sophisticated semantic datasets and by recent advances in deep language processing. Basque is one of the 8 languages (Basque, Bulgarian, Czech, Dutch, English, German, Portuguese and Spanish) involved in the project, and the tools presented in this work are being used for its linguistic processing.

The other project in which these tools have been used is Ber2tek,[11] whose aim is to advance in the research and development of the technologies of analysis of cross-media contents, high quality machine translation and natural spoken multimodal interaction for Basque.

6 Conclusions and Future Work

Many other multilingual NLP toolkits exist, but integrating already developed tools into most of them is not trivial. In this paper, we have presented our approach to adapt and integrate a set of NLP tools for Basque using a modular architecture.

The main characteristics of such tools are the following: to allow the integration of complex tools easily, to transmit information containing morphologically rich annotations among tools, and to be modular.

The robustness of the modular chain of Basque processing tools is already being tested doing extensive processing, mainly in the FP7 European project QTLeap.

Currently we have made publicly available two tools of the chain, namely, the morphosyntactic analyzer and PoS tagger, and the dependency parser. Moreover, tools such as semantic role labeling, named entity disambiguation and coreference resolution are being integrated and will be available soon.

[10] `http://qtleap.eu`
[11] `http://www.ber2tek.eus/en`

Acknowledgments

This work has received support by the EC's FP7 (FP7/2007-2013) under grant agreement number 610516: "QTLeap: Quality Translation by Deep Language Engineering Approaches".

References

1. Aduriz, I., Agirre, E., Aldezabal, I., Alegria, I., Arregi, X., Jose, M.A., Artola, X., Gojenola, K., Sarasola, K., Urkia, M.: A word-grammar based morphological analyzer for agglutinative languages. In: Proceedings of COLING. pp. 1–7 (2000)
2. Aduriz, I., Aranzabe, M., Jose, M.A., Atutxa, A., de Ilarraza, A.D., Ezeiza, N., Gojenola, K., Oronoz, M., Soroa, A., Urizar, R.: Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for the automatic processing, Corpus Linguistics Around the World. Language and Computers Series, vol. 56, pp. 1–15 (2006)
3. Aduriz, I., et al.: Construction of a basque dependency treebank. In: Proceedings of the Workshop on Treebanks an Linguistic Theories (TLT03)
4. Agerri, R., Bermudez, J., Rigau, G.: IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In: Proceedings of LREC 2014. pp. 3823–3828 (2014)
5. Agerri, R., Rigau, G.: Robust multilingual Named Entity Recognition with shallow semi-supervised features. Artificial Intelligence 238, 63 – 82 (2016)
6. Aranzabe, M., Atutxa, A., Bengoetxea, K., de Ilarraza, A.D., Goenaga, I., Gojenola, K., Uria, L.: Automatic conversion of the basque dependency treebank to universal dependencies. In: Proceedings of the Workshop on Treebanks an Linguistic Theories (TLT14). pp. 233–241 (2015)
7. Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36(2), 223–254 (2002)
8. Ezeiza, N., Aduriz, I., Alegria, I., Arriola, M., Urizar, R.: Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. In: Proceedings of COLING-ACL'98 (1998)
9. Fokkens, A., Soroa, A., Beloki, Z., Ockeloen, N., Rigau, G., van Hage, W.R., Vossen, P.: NAF and GAF: Linking linguistic annotations. In: Proceedings 10th Joint ISO-ACL SIGSEM Workshop on Interoperable Semantic Annotation (2014)
10. Goenaga, I., Gojenola, K., Ezeiza, N.: Exploiting the contribution of morphological information to parsing: the BASQUE TEAM system in the SPRML'2013 shared task. In: Proceedings of SPRML-2013 Workshop, ACL. pp. 71–77 (2013)
11. Karlsson, F., Voutilainen, A., Heikkila, J., Anttila, A. (eds.): Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text (1995)
12. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of ACL 2014: System Demonstrations. pp. 55–60 (2014)
13. Padró, L., Stanilovsky, E.: FreeLing 3.0: Towards Wider Multilinguality. In: Proceedings of LREC 2012 (2012)