Top-Performing Robust Constituency Parsing of Portuguese: freely available in as many ways as you can get it

João Silva, António Branco, Patricia Gonçalves

University of Lisbon
Edifício C6, Departamento de Informática
Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal
{jsilva, antonio.branco, patricia.nunes}@di.fc.ul.pt

Abstract

In this paper we present LX-Parser, a probabilistic, robust constituency parser for Portuguese. This parser achieves ca. 88% f-score in the labeled bracketing task, thus reaching a state-of-the-art performance score that is in line with those that are currently obtained by top-ranking parsers for English, the most studied natural language. To the best of our knowledge, LX-Parser is the first state-of-the-art, robust constituency parser for Portuguese that is made freely available. This parser is being distributed in a variety of ways, each suited for a different type of usage. More specifically, LX-Parser is being made available (i) as a downloadable, stand-alone parsing tool that can be run locally by its users; (ii) as a Web service that exposes an interface that can be invoked remotely and transparently by client applications; and finally (iii) as an on-line parsing service, aimed at human users, that can be accessed through any common Web browser.

1. Introduction

Parsing is a central task in Natural Language Processing, necessary for any procedure that needs to work with the grammatical structure of a sentence.

When opting for a hand-built parser, one will have to invest a great deal of effort into building the grammar rules and lexicon. If, in turn, one opts for a trainable parser, there is still need to annotate a training corpus, which is in itself an onerous and time-consuming task. Given this situation, the present release of a freely usable parser is expected to be very helpful for those that do not have access to such a tool.

In this paper, we present LX-Parser, a probabilistic, robust constituency parser for Portuguese. It achieves state-of-the-art performance scores—ca. 88% f-score for labeled bracketing—on a par with the scores currently achieved for the most studied natural language, English.

This parser is, to the best of our knowledge, the first freely available, robust constituency parser for Portuguese.

LX-Parser is being distributed in a variety of ways: (i) as a stand-alone parsing tool that can be downloaded and ran locally by its users; (ii) as a Web service whose API can be invoked remotely by client applications; and finally (iii) as an on-line parsing service, aimed at human users, that can be accessed through any common Web browser.

This parser, and the different modes under which it is made available, are described in greater detail in the following sections.

2. Standalone parser

Differently from parsers with hand-built lexica and rules—which eventually fail to deliver a result for input sentences containing words or syntactic construction not covered by their underlying grammar—robust parsers are designed to always deliver a parse to any input sentence, even if at the cost of only being able to achieve sub-optimal accuracy.

The most widely-used technology underlying robust constituency parsers is that of probabilistic context-free grammars (PCFG). This technology has now matured to a point where it is able to consistently achieve 85–90% f-score for the task of labeled constituency analysis (Petrov et al., 2006, p. 440).

Due to the effort put into this area over the past few years, there is currently a wide range of language-independent, freely available software packages that allow training a robust probabilistic parser.

The experiment reported in (Silva et al., 2010) studies several such packages in order to assess: (i) their suitability for out-of-the-box training of a robust parser for Portuguese; and (ii) to what extent does the resulting parser perform at state-of-the-art levels.

To do this, the different software packages were run over a treebank with $1,204$ sentences ($10,387$ tokens), mostly formed by newspaper articles. This treebank was produced from the output of a deep processing linguistic grammar by letting trained linguists select the correct parse from among all possible parses that were delivered by the grammar for a given sentence.

Parser evaluation was performed using a standard 10-fold cross-validation methodology where 90% of the dataset was used for training the parser and the remaining 10% for evaluating it.

The robust parsers that were obtained were able to achieve f-scores in the 85–89% range for labeled constituency analysis. That is, they fall within the window of scores being currently achieved for the English language by top-scoring probabilistic parsers (Petrov et al., 2006, p. 440).

It is important to note that the software packages used to create the parsers described in (Silva et al., 2010) were run out-of-the-box and with their default parameters. That is, the various constituency parsers that where obtained were in no way specifically tuned or adapted to the Portuguese language. Nevertheless, all parsers achieved state-of-the-art scores and thus were deemed to be—in terms of accuracy—suited to support LX-Parser.

The services that are discussed in this paper are currently being supported by the Stanford parser (Klein and Manning, 2003). This is a factored parser, which models phrase-structure and lexical dependencies separately as independent models. The *rationale* being that, by separately modeling this information, each model will be simpler, more efficient and less sensitive to data-sparseness problems.

Phrase-structure is modeled using a standard PCFG, where node labels have been enriched with a variety of contextual features, such as parent annotation of nodes, in a manner similar to the one described in (Collins, 1999).

Lexical dependencies are modeled by taking into account a variety of features, such as distance and valence, between the head of a constituent and its dependents.

The overall probability that the parser assigns to a parse tree is then given by the product of the probabilities that each of the two models independently assigns to that tree (Klein and Manning, 2003).

The motivation for choosing the Stanford parser to support the services described here is twofold. Firstly, this the second best scoring parser found in the study reported in (Silva et al., 2010), where it scored 88% f-score in the labeled bracketing task, very closely following the top-ranking one. Additionally, it is the easiest to integrate into other applications due to its design and well-documented implementation.

Release. The parser, as a stand-alone tool, is made available at `http://lxparser.di.fc.ul.pt/`. Note that it is only the learned language model (i.e. the so-called "grammar" file) that is being made available here. The parsing engine itself—a Java application—should be downloaded separately as part of the freely available Stanford Parser software package.[1]

3. Web service

A Web service can be broadly defined as way of exposing the application programming interface (API) of an application in such a way that it is made accessible to other applications over the Web (Alonso et al., 2004, ch. 5).

When choosing to provide access to the parser as a Web service, we follow the same *rationale* as when creating LXService, a Web service providing a set of tools for shallow processing of Portuguese that includes, among other tools, a part-of-speech tagger with state-of-the-art performance (Branco et al., 2008).

To wit, managing the distribution of language resources can involve several non-trivial issues ranging from intellectual property concerns to maintaining users up-to-date with newer versions of the resource.

Recent initiatives, such as the pan-European CLARIN project,[2] seek to address these issues by distributing language resources via Web services (Váradi et al., 2008).

Such an approach allows not only humans to access those resources but also computer applications, while authentication and authorization concerns can be handled in an automatic and transparent manner.

This approach to the distribution of language resources can similarly be applied to the distribution of language technology. Having this kind of seamless access to a variety of language technologies and resources will leverage strengths and perhaps lead to a widespread incorporation of language technology into a range of different applications.

To those that wish to use the parsing Web service, a username and password are granted. These are subsequently used by the client application to authenticate itself before the server.

Data transfer is supported by the Simple Access Object Protocol (SOAP), namely through the Apache Axis implementation. Accordingly, the relevant Apache Axis libraries must be installed on the client.[3]

The API that is made accessible to the client application contains a `parse` method. This method accepts a plain text sentence as a Java `String` and returns its parse tree, represented in a plain text bracketed format, also as a Java `String`.

This particular choice of output format is motivated by the fact that this is the same format that is used in the Penn Treebank and, due to that, has become the *de facto* standard for textually representing this type of trees. An example of the input and output formats, together with a graphical representation of the latter, can be seen in Figure 1.

As the example shows, input to the Web service is a plain text sentence that does not need to be previously tokenized. In particular, punctuation symbols are still adjacent to words and contracted forms, such as `No`,[4] are still represented as a single token. The tokenization of the input is performed server-side, before running the underlying parser, by the same shallow processing tools used to power LXService.

This is worth of note since tokenization is a non-trivial task, mainly due to the existence of token-ambiguous strings. That is, strings that can be tokenized either into one or into two tokens, depending on the part-of-speech they bear in a particular occurrence (Silva, 2007, p. 19).

For instance, the word `deste` is to be tokenized as a single token if it is a form of the verb `dar`, but as two tokens if it is an occurrence of the contraction of the Preposition `de` and the Demonstrative `este`.

The tokenizer that is part of LXService—and used to preprocess the input to LX-Parser—is able to resolve these ambiguous strings with 99.44% accuracy (Silva, 2007).

The present API, which for now contains the `parse` method, is currently being extended to provide additional methods and, in particular, access to a greater variety of parsing options, such as assigning syntactic functions to constituents.

An interesting and powerful synergy that arises from the several Web services that have been made available is that LX-Parser can be coupled with LXService, the latter providing shallow pre-processing of the input, such as part-of-speech tagging or lemmatization, which can help improve parser performance.

[1]Stanford Parser package: `http://nlp.stanford.edu/software/lex-parser.shtml`.

[2]Website at `http://www.clarin.eu/`.

[3]Website at `http://ws.apache.org/axis/`.

[4]The contracted form `no` expands into two tokens: the Preposition `em` and the Definite Article `o`.

Input format:

```
No Porto, as coisas não estão
melhores.
```

Return format:

```
(S (S (PP (PP (P Em) (NP (D o)
(N Porto))) (PNT ,)) (S (NP (D as)
(N coisas)) (VP (V′ (ADV não)
(V estão)) (A melhores)))) (PNT .))
```

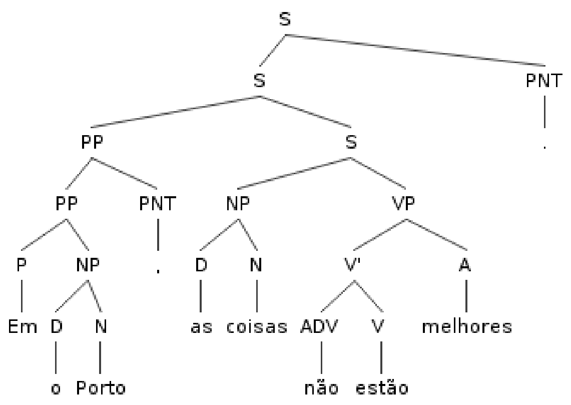


Figure 1: Input and output formats for the `parse` method

More importantly, by virtue of the very nature of a Web service, any improvement that is made to the service—such as exposing a richer API, or providing the underlying parser with a new model trained over a larger treebank—will immediately and seamlessly be available to all client applications.[5]

Release. To ease implementation, a Java package is provided that, when imported by an application, gives access to the methods (e.g. `parse`) needed for that application to function as a client to LX-Parser. This package is available at `http://lxparser.di.fc.ul.pt/`.

4. On-line parsing

Providing access to the parser through a Web service does not preclude us from also presenting the same parsing service under a more friendly, human-oriented interface.

With this in mind, the parser is also used to support an online service which allows the user to access a website where a plain text sentence is entered and an image of its parse tree is shown as a result.

The sentence that the user types in is tokenized and passed on to the parsing service. The resulting parse tree is shown in the browser window through the use of the `phpSyntaxTree` package.[6] The image of the parse tree can be saved in Portable Network Graphics (PNG) and Scalable Vector Graphics (SVG) formats.

The options that are available to the user of this on-line parsing service go hand-in-hand with the capabilities of its sister Web service.

The website for the on-line service can be reached at `http://lxparser.di.fc.ul.pt/`. A screenshot of this webpage is shown in Figure 2.

5. Concluding remarks

In this paper we presented a constituency parser that is, to the best of our knowledge, the first freely available robust probabilistic parser for Portuguese. It achieves state-of-the-art performance with 88% f-score for labeled bracketing.

This parser is distributed in a variety of ways. It is made available not only as a downloadable stand-alone tool, but also as a Web service that can be seamlessly used by client applications. The parser is also used to support a website where the user can enter a plain text sentence and get an image of the resulting parse tree.

From this point onwards, work will be directed along two main axes.

On the one hand, we will improve the underlying parser by fine-tuning its parameters and by training over a larger treebank. This is a promising approach to raise the accuracy of the parser since extremely good results were achieved even when training over a small treebank and without any effort to specifically adjust the parser to Portuguese—for instance, by creating rules for head-finding.

On the other hand, we will improve the Web service. This will be done by standardizing the access to it by including it into Universal Description, Discovery and Integration (UDDI) directories. Also by enriching its API so that users can choose whether constituents should also be assigned syntactic function tags, and whether input should also be processed by the shallow processing tools in LXService—e.g. a POS-tagger, a lemmatizer or a named-entity recognizer—before being parsed.

6. References

Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. 2004. *Web Services: Concepts, Architectures and Applications*. Springer Verlag.

António Branco, Francisco Costa, Filipe Nunes, João Silva, and Sara Silveira. 2008. LXService: Web services of language technology for Portuguese. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC)*, pages 2577–2583.

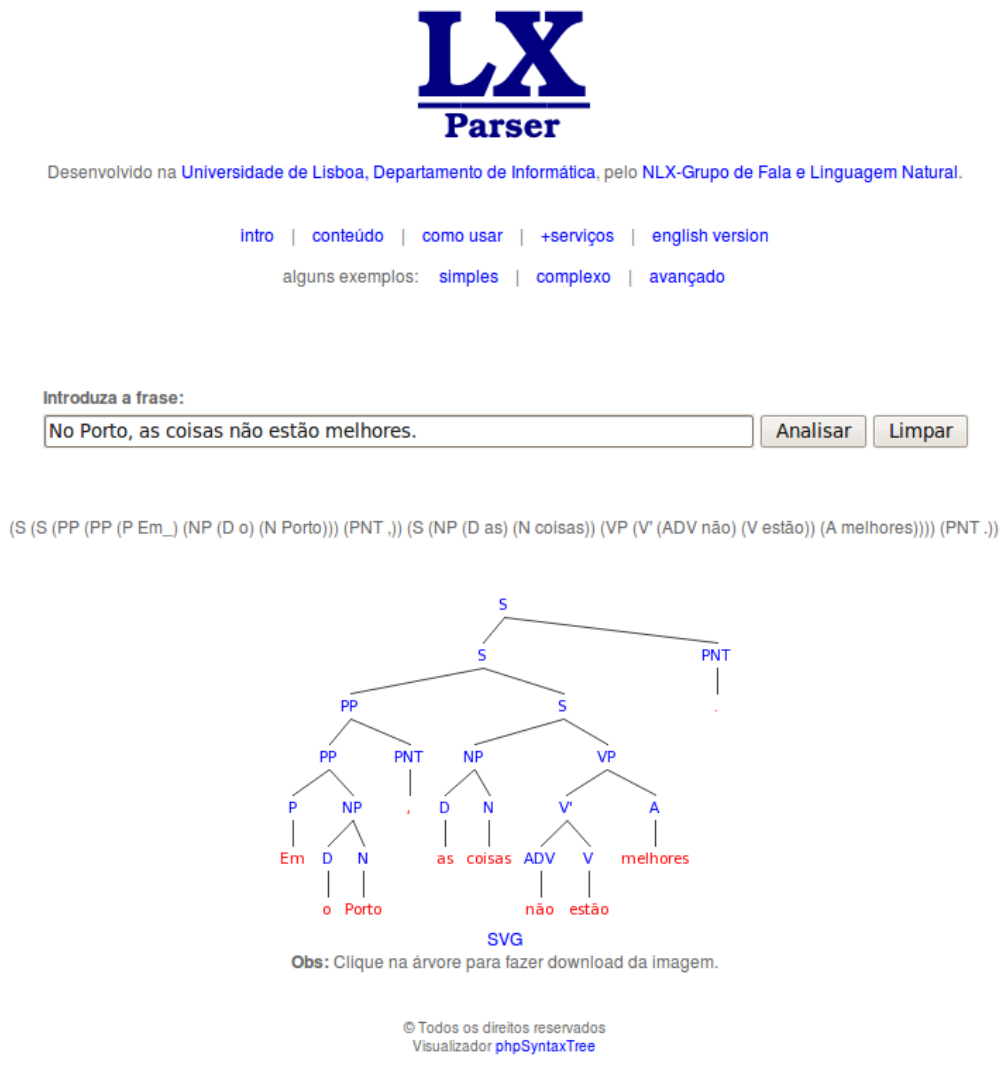
Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Dan Klein and Christopher Manning. 2003. Fast exact inference with a factored model for NLP. *Advances in Neural Language Processing Systems*, 15:3–10.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting*

[5]Access to previous versions of the parser can be maintained by extending the API. For instance, the `parse` method could be overloaded to accept a version identifier as an optional argument.

[6]The website for the tree viewer software can be found at `http://ironcreek.net/phpsyntaxtree/`.


Desenvolvido na Universidade de Lisboa, Departamento de Informática, pelo NLX-Grupo de Fala e Linguagem Natural.

intro | conteúdo | como usar | +serviços | english version

alguns exemplos: simples | complexo | avançado

Introduza a frase:

No Porto, as coisas não estão melhores.

Analisar Limpar

```
(S (S (PP (PP (P Em_) (NP (D o) (N Porto))) (PNT ,)) (S (NP (D as) (N coisas)) (VP (V' (ADV não) (V estão)) (A melhores)))) (PNT .))
```

SVG

Obs: Clique na árvore para fazer download da imagem.

© Todos os direitos reservados
Visualizador phpSyntaxTree

Figure 2: Screenshot of the on-line service

of the Association for Computational Linguistics (ACL), pages 433–440.

João Silva, António Branco, Sérgio Castro, and Ruben Reis. 2010. Out-of-the-box robust parsing of Portuguese. In *Proceedings of the 9th Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR)*.

João Silva. 2007. Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, University of Lisbon. Published as Technical Report DI-FCUL-TR-07-16 at `http://hdl.handle.net/10455/3095`.

Tamás Váradi, Steven Krauwer, Peter Wittenburg, Martin Wynne, and Kimmo Koskenniemi. 2008. CLARIN: Common language resources and technology infrastructure. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC)*, pages 1244–1248.