Adaptation of Data and Models for Probabilistic Parsing of Portuguese

Benjamin Wing and Jason Baldridge

Department of Linguistics,
University of Texas at Austin,
Austin TX 78712, USA
{benwing, jbaldrid}@mail.utexas.edu

Abstract. We present the first results for recovering word-word dependencies from a probabilistic parser for Portuguese trained on and evaluated against human annotated syntactic analyses. We use the Floresta Sintá(c)tica with the Bikel multi-lingual parsing engine and evaluate performance on both PARSEVAL and unlabeled dependencies. We explore several configurations, both in terms of parameterizing the parser and in terms of enhancements to the trees used for training the parser. Our best configuration achieves 80.6% dependency accuracy on unseen test material, well above adjacency baselines and on par with previous results for unlabeled dependencies.

1 Introduction

Early work on probabilistic parsing focused primarily on English; there is now a growing body of work regarding building treebanks and parsers for other languages. [1] performed one of the first cross-linguistic probabilistic parsing experiments, using the Czech Prague Dependency Treebank [2]. They converted the dependency representations in the treebank to tree structures and then trained various head-driven parsing models [3]. More recent work includes probabilistic parsing for German [4, 5] and French [6].

Portuguese presents many challenges for parsing. Although its nominal inflections are somewhat simpler than languages like Czech and its word order is more restricted, its verbal inflections are significantly more complex. Verbs are conjugated in six person-number combinations and ten synthetic tenses, as well as various non-finite forms. Verbs are lexicalized into three declensional families, and there are numerous subclasses and irregularities. In addition, many verbal endings are identical to inflectional or derivational suffixes used to form nouns, significantly complicating the task of morphological analysis.

A previous statistical parser for *historical* Portuguese, using the Tycho Brahe corpus, was developed by [7]. Using roughly 2000 human-annotated sentences, PARSEVAL f-scores in the 51% to 56% range were obtained with two fairly simple statistical models. A standard Collins parser [8] was implemented by [9, 10] and trained using the CetenFolha corpus (see section 2). However, no manual annotation was then available for this corpus. As a result, the parser

R. Vieira et al. (Eds.): PROPOR 2006, LNAI 3960, pp. 140–149, 2006.

was evaluated only qualitatively, on 23 sentences annotated by the author; it is unclear whether these results can be generalized.

There now exists a substantial corpus of Portuguese texts annotated with quasi-dependency structures, the Floresta Sintá(c)tica [11, 12]. Like the corpus used by [10], the analyses are based on the output of the PALAVRAS parser, but for the Floresta, they have been hand-corrected by human annotators to create a gold standard corpus of analyses. However, this resource has until now not been used to train probabilistic parsers for Portuguese.

In this paper, we describe head-driven generative probabilistic parsing models for Portuguese using the Floresta and the Bikel multi-lingual parsing engine [13, 14]. We evaluate parsing performance, using both standard PARSEVAL and unlabeled dependency accuracy, for differing levels of effort in adapting the parser for Portuguese data and adapting the data for the parser. We show that making relatively straightforward changes to the data itself and the parameterization of Bikel's parser – including sensitivity to Portuguese morphology – pays large dividends in performance. Our best model achieves 81.0% unlabeled dependency accuracy and 63.2% PARSEVAL f-score on unseen test material. In section 2, we discuss the Floresta and its properties. The next section describes how we produce training material from the Floresta in the appropriate format for the parser and make augmentations to the resulting trees to improve their training utility for the parser. Section 4 introduces the parsing model we use and how we modify it for parsing Portuguese. Section 5 describes how we run our parsing experiments and reports the performance of the various configurations we tested. The last section concludes and describes future work.

2 The Floresta Sintá(c)tica

The Floresta Sintá(c)tica consists of 9,374 sentences and 214,490 tokens, split into two parts of approximately equal size. One part, the CetenFolha (CF), consists of 4,213 sentences and 80,015 tokens taken from the Brazilian newspaper *Folha de São Paulo*. The other, the CetemPúblico (CP), consists of 5,161 sentences and 134,475 tokens taken from the Portuguese newspaper *Público*. The syntactic annotations were produced by hand-correcting the output of the PALAVRAS parser ([15]), a non-statistical parser containing a 75,000-word lexicon and a 2,000-line grammar of inflectional and derivational rules.

Sentence (1) is an example from the corpus. Its analysis is given in Figure 1.

(1) Veículos de resgate estavam a apenas 500 metros do Airbus 300.
Vehicles of rescue were at just 500 meters from.the Airbus 300.
Rescue vehicles were only 500 meters from the Airbus 300.

The annotations provide full morphological analyses of each word and syntactic analyses of each sentence. Each word has a functional tag and part-of-speech tag. `H:n`, for example, specifies a functional tag `H` (head) and a POS tag `n` (noun). `STA:fcl` indicates a finite verbal clause (`fcl`) that is a statement (`STA`). Morphological information for each word is given as the word's lemma and a set

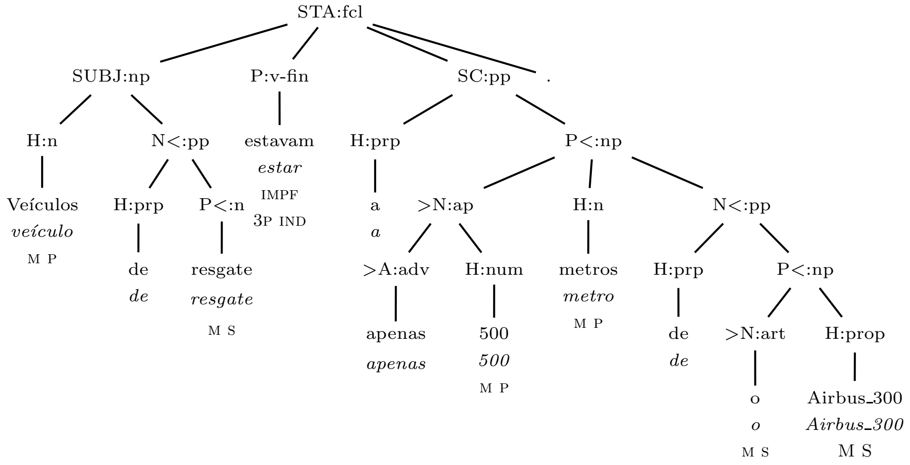

Fig. 1. Floresta tree analysis for sentence (1), from the *arvores deitadas* format

of grammatical features. For example, the annotation "*metro* M P" for *metros* "meters" indicates that it is masculine (M) and plural (P) and its lemma is *metro*.

The tokenization of the text in the syntactic annotations differs quite radically from that of the raw text. The corpus consistently splits combinations of prepositions and determiners, while many named entities and multi-word expressions are joined as one token. In (1), for example, *do* "of the" is split into *de* and *o*, and *Airbus 300* is joined as *Airbus_300*.

The Floresta explicitly indicates heads, arguments and non-argument modifiers. Heads are generally marked with a functional tag of H, P or MV, depending on the constituent type. Arguments are indicated using functional tags such as SUBJ (subject), ACC (direct object), PIV (prepositional object), and SC (subject clause). Non-argument modifiers are indicated using functional tags containing a > or <, indicating a pre-modifier and post-modifier, respectively. The analysis in Figure 1 contains examples of nominal modifiers (N< and >N), prepositional modifiers (P<), and adverbial modifiers (A<).

3 Preparing the Training Material

In order to use the Floresta as training material for the Bikel parser, we converted it from its native format into the format used for the Penn Treebank (PTB). We did so in two ways. One is as straightforward as possible and involves no modification to the constituent labels or structures as represented in the *arvores deitadas* (AD) format. The second makes minor changes to the trees and their labels in order to improve their utility for head-driven parsing models.

Certain aspects of the AD format make a complete one-to-one mapping to Penn Treebank format impossible. As a result, we include three transformations to create PTB-style trees. First, punctuation is simply listed as-is in the AD

```
if any children have an AUX functional tag,
  then the leftmost one of these is the head
else if any children have a head marker functional tag (H, P, MV, PMV or PAUX),
  then the leftmost one of these is the head
else if the constituent is a conjunction, then the leftmost conjunct is the head
else if (the label of N is acl (adverbial clause) and
        any children have functional tag COM (complementizer) or PRD (predicator)),
    then the leftmost of these is the head
else if a child has POS tag cu and functional tag ?, then it is the head
else
  collect the set C of children that are neither punctuation nor have a functional
  tag that indicates a non-head
  if C is non-empty
    if N is the root of the tree, then the leftmost of these children is the head
    else choose the head from them in the order:
      clause, conjunction, noun phrase, the leftmost one
  else the leftmost child is the head
```

Fig. 2. Test for determining the head of a node N

format, but requires an associated tag in PTB format. For this, we add tags consistent with Penn Treebank usage; e.g. "`.`", "`?`" and "`!`", are tagged with "`.`". Second, the AD format includes some types of information that cannot be encoded in PTB format, such as morphological analyses and declarations of multiple possible attachment points for some constituents. We simply delete this information as part of the conversion. Finally, the Floresta has discontinuous constituents, which we map into separate constituents.[1]

These simple transformations provide a baseline set of trees that can be used to train a parser. However, it is often the case that steps can be taken to massage the trees in a treebank to improve the parameterization of the parsing models [1, 8, 14]. For the Floresta, we do three main augmentations to the trees: (a) adding explicit head markers, (b) improving the representation of conjunctions, and (c) distinguishing relative clause nodes from other clause level nodes.

Information regarding the heads of constituents in trees is fundamental for deriving dependency relations from treebanks and for parameterizing our parsing models. The PTB format does not mark heads explicitly, so head-driven parsers typically use a complex set of heuristics to determine the head of each constituent. However, heads are (usually) marked explicitly in the Floresta, so we use this to indicate the heads in PTB format by adding `/H` to their label. There are some cases where heads are not marked explicitly – our full test for determining the head of a constituent is given in Figure 2.

To derive the dependency relations, we mostly just create a link from the head word of each non-head child of a constituent to the head word of the constituent's head child. However, to be as consistent as possible with the Portuguese data

[1] There were other minor formatting issues in the conversion, such as standardizing open and closed quotation marks.

in the CoNLL-X shared task on dependency parsing, we need more complex handling of verbal groups (a constituent in the Floresta consisting of a main verb and any corresponding auxiliaries). Verbal subjects (approximated by choosing constituents to the left of a verb that do not have an adverbial tag, i.e. /ACL, /ADV, /ADVP or /PP), as well as all punctuation, are dependents of the first auxiliary, but all other constituents are dependents of the main verb. In addition, each verb in the verbal group is dependent on the verb to its left.

Another change we make to tree labels improves the representation of conjunctions. Conjoined clauses in the native Floresta are of type CU, regardless of the type of constituents being conjoined. This causes grammars learned from the treebank to make errors such as conflating noun phrase conjunctions and sentential conjunctions. We thus augment the syntactic type of conjuncts to include the type of the conjoined constituents by using the syntactic type of the head child. This is similar to what was done for Czech by Collins et al. [1].

Following another transformation given in [1], we augment clauses under NPs to distinguish relative clauses from clauses in other circumstances. Essentially, this creates a distinction between a "clause" and a "clause-bar". We identify such clauses by looking for `acl`, `icl` and `fcl` children of `np` constituents.

4 Adapting the Parser for Portuguese

We use Bikel's multi-lingual parsing engine [13, 14] to train and run parsing models for Portuguese. The parser implements and extends the parsing models of Collins [8], which include several lexicalized head-driven generative parsing models that incorporate varying levels of structural information, such as distance features, the complement/adjunct distinction, subcategorization and gaps.

The parsing model we use is essentially Collins' model 2, with the addition of the first-order bigram dependencies described in [1]. With this extension, the generation of a modifier is also dependent on the previous modifier:

$$\prod_{i=1\ldots n+1} \mathcal{P}_l(L_i(l_i)|L_{i-1}, P, h, H)$$

We use Bikel's default approximation of the previous modifier. It is either the (a) START symbol (no previous modifiers), (b) a coordinating conjunction, (c) a punctuation mark, or (d) MISC for all other modifiers.

The Bikel parser allows language-specific extensions to be created. It comes out-of-the-box with support for English, Arabic and Chinese. In addition to using the English package to determine a baseline parsing accuracy, we created a package for Portuguese. This package provides head-finding rules, special handling for when heads are explicitly marked, morphological features, argument/non-argument marking, and some tuning of parser options for the Floresta.

Head-driven parsing models must know the head child of each constituent during training. This information is not encoded in the PTB, so the English package provides a series of head-finding heuristics. For each constituent type, an ordered list of syntactic types is given; the parser searches in turn for a child of each type, assigning the head to the first such child found. For the Portuguese package, we modified these rules as appropriate for the Floresta. We also modified the parser to be aware of the explicit `/H` head indications, as described in Figure 2.[2] When these indications are present, they are marked for every constituent, and thus the head-finding rules are unused. However, the parser will fall back onto these rules as necessary, as in our baseline Portuguese model.

Each language package also can encode features based on morphological properties of a word – these are especially important for unknown words. Five types of features are encoded for each word: capitalization, hyphenation, numeric, inflection, and derivation. The first three indicate, respectively, whether words are capitalized, contain hyphens, or are in the form of numbers. For the most part the code to create them needed no changes. We extensively modified the latter two, however, to handle the morphology of Portuguese.

The inflectional and derivational features indicate the presence of particular suffixes in a word. We created a list of 39 of the recognizably nominal or verbal inflectional endings in Portuguese. This required some care to avoid hitting false positives while at the same time avoiding spreading the features too thin. Thus, we have a single *-rem* to handle the various 3rd plural future subjunctive endings, but separate *-ado* and *-ido* to avoid false positives on nouns like "caldo" and "medo". Furthermore, some endings are not listed at all (e.g. *-o*, *-a*) because they are too ambiguous and are not reliably nominal or verbal. We also modify the handling of plural *-s*; Portuguese plurals nearly always involve a vowel followed by an *-s*, whereas English plurals can have *-s* after various consonants.

We list a series of stop words that should not be segmented. This includes collocations formed by joining multiple words together – these are largely proper names. It also includes words in *-gem* (confusable with verbal *-em*) and a series of common words for the various endings. (E.g. *lugar*, *mar*, *popular* for verbal *-ar*; *classe*, *esse*, *interesse* for verbal *-sse*; *quer*, *qualquer*, *mulher* for verbal *-er*).

We likewise made extensive modifications for the derivational features. We list all common derivational features that are not easily confusable with inflectional features or that rarely occur as inflections. (For example, *-ara* is a literary pluperfect verbal form as well as a nominal ending, but the pluperfect rarely occurs.) We also have special code to handle plurals of suffixes that end in a vowel, without the need to explicitly list each such plural form.

For Collins' model 2, the parser needs to be able to distinguish arguments and non-arguments during training. We found that the heuristic rules used for handling the PTB could be adapted without major work to handle the Floresta as well, since they make explicit reference to the functional tags. Although the PTB, unlike the Floresta, does not explicitly indicate arguments, it does include functional tags of various sorts that are identifiably *not* arguments: these are what are listed in the heuristic rules. We could in principle change how these rules worked for the Floresta, but in practice it worked well to follow the same format

[2] This augmentation is removed at the end of preprocessing to avoid encoding it in the parsing model. This would create difficulties when using tags suggested by a tagger.

and list those functional tags that are clearly not arguments. The remaining nodes are identified as arguments when they occur in the appropriate contexts (e.g., a nominal or clausal child of a clausal constituent). Thus, it sufficed simply to enumerate syntactic tags that identify nominal and clausal constituents and functional tags that cannot be arguments (i.e. modifiers, adverbials and the like).

We made other minor changes to the parser settings. For example, we use Knesser-Ney smoothing instead of the default Witten-Bell, we use an unknown word threshold of two rather than six, and we turn off a number of options that are quite specific to PTB trees. Finally, we restrict the parser so that it makes no unary productions.

5 Experiments

We consider three different parser/data configurations that vary the amount of effort put into adapting the base to the Floresta: BAS-ENG – basic trees with the standard English language package; BAS-PORT – basic trees with the Portuguese package; and AUG-PORT– augmented trees with the Portuguese package. The first represents the laziest approach: do nothing other than ensuring that the trees can be used by the parser. The second makes the parser aware of the language/corpus, while the third involves changing the trees themselves to be more informative to the parser, as described in section 3. We use three different sources of part-of-speech tags: tags obtained from the parser itself (PTAGS), from a tagger[3] (TTAGS), and from the Floresta itself (GTAGS). The latter is used only to show an upperbound on parser performance for each configuration.

We evaluate performance both in terms of standard PARSEVAL *f*-scores[4] and unlabeled word-word dependencies. We derive our gold standard dependencies as described in section 3. PARSEVAL is a useful way of seeing how well the trees themselves are being modeled by the parser, but the dependency accuracy is the true evaluation. It provides a more clear indication of whether the fundamental relationships recorded in the Floresta are being recovered.

For our experiments, we created a development/training set and test set by randomly sampling from the sentences in the Floresta. The development set has 7497 sentences with 170,527 dependency links, and the test set has 1877 sentences with 42,254 dependency links. We refined our models/configurations using 10-fold cross validation on the development set, and give the performance of our best configuration on the test set.[5]

Figure 3 shows the PARSEVAL *f*-scores and dependency accuracies for the various configurations. The BAS-ENG configuration unsurprisingly has the worst performance. Though it is not entirely random, we see that simply putting in the relatively minimal effort to create the Portuguese language package leads to large 20-25% absolute improvements in performance in the BAS-PORT configuration. For example, compare PTAGS *f*-score of 36.3% for BAS-ENG to 60.9% for BAS-

[3] We use the OpenNLP Toolkit maxent tagger, available from `opennlp.sf.net`.

[4] The *f*-score is calculated as $\frac{2 \times precision \times recall}{precision + recall}$.

[5] We will make the sentence ids in the two sets available to facilitate future comparison.

Model	BAS-ENG			BAS-PORT			AUG-PORT		
	PTAGS	TTAGS	GTAGS	PTAGS	TTAGS	GTAGS	PTAGS	TTAGS	GTAGS
F-score	36.3	37.0	38.6	60.9	60.6	63.8	63.2	63.2	67.1
Dependency Acc.	17.4	18.0	18.4	72.8	73.2	75.7	80.7	81.0	84.0

Fig. 3. PARSEVAL f-scores and dependency accuracy for 10-fold cross-validation experiments with development material

PORT. The transformations to the trees in the AUG-PORT configuration – explicit heads, finer-grained coordination labels, and distinguishing relative clauses – produce a smaller, but significant 2-3% absolute improvement in performance.

The different tagging configurations show that using gold tags results in the best performance. Also, despite the fact that that the tagger tags more accurately than the parser (96.0% vs 94.1%), there is no significant difference in performance between PTAGS and TTAGS for any of the configurations. This is consistent with what was found for Czech [1]. In the TTAGS configuration, we trained the parser on gold standard tags, but tested it with tags from the tagger. Even though the tagger's suggestions are less accurate than the gold standard tags, it can be actually beneficial to use its output in the *training* trees [1]. That way, the tags in training are more like those that the parser will see on tagger-tagged test material. Regardless of how a tagger affects performance, it does have the benefit of speeding up parsing considerably.

The dependency scores in Figure 3 show a similar pattern to the PARSEVAL scores, apart from the BAS-ENG configuration. These scores are compared against left and right linking baselines (i.e., words are dependents of the word to their left or right), which are 26.8% and 22.6%, respectively. The dependency scores for BAS-ENG are worse than either baseline and are relatively much lower than they were for PARSEVAL. This is essentially due to the complete lack of head information, which means that the dependencies extracted from the BAS-ENG parser output are often incorrect because the wrong head was chosen by the parser. PARSEVAL does not reflect this since it only concerns the label and span of a constituent, not the relationships between its children.

The head information provided by the head heuristics in our Portuguese language package is the most likely influence in the considerably better performance of the BAS-PORT configuration, which overwhelmingly improves on both the BAS-ENG configuration and both baselines. When adapting a parser such as Bikel's to a new language, it clearly pays to put in the minimal effort to write even a rough set of reasonably accurate head finding rules.

The even more explicit handling of heads and the tree improvements together then provide a large 8% absolute improvement for AUG-PORT. It is easy to see why the change to coordination labels can make a big difference in the discriminitive capabilities of the parsing model. It makes predictions mostly based on the relationship between children and parent nodes rather than between grandparents and grandchildren. It thus cannot see beyond a simple CU node to know that it contains two NP conjuncts and thereby determine whether they together make a good argument for a verb. The change also prevents coordination of unlike constituents [1]. Distinguishing relative clauses improves the handling of subcategorization of different types of clause level constituents since relative clauses nearly always lack one of the arguments of the verb. Also, they should not be coordinated as a like type with other clauses.

Our best configuration on the development material is AUG-PORT, with no significant difference between using PTAGS or TTAGS. The performance AUG-PORT-PTAGS on the 1877 sentences in the *test* set is an *f*-score of 63.8% (64.7% precision, 62.9% recall) and unlabeled dependency accuracy of 79.9%. For AUG-PORT-TTAGS, we obtain *f*-score of 63.3% (64.1% precision, 62.5% recall) and dependency score of 80.6%. Both dependency scores overwhelmingly beat left and right linking baselines on the test material of 26.9% and 22.6%, respectively, and they are on par with the results obtained for Czech.

We also performed a basic error analysis, investigating the 60 sentences with between 10 to 20 words with the worst dependency figures. The largest source of error was coordination problems (58%), especially in the presence of multiple elements (57% of the coordination problems). The second major source was relativization problems (28%). Some of the additional issues were incorrect handling of subordination (20%), overly eager creation of verb groups (12%), and difficulties handling quoted sentences (12%), fragments (8%), and non-NP subjects (8%). 13% of the sentences revealed errors in the Floresta. This analysis largely vindicates the input transformations we chose. It also points the way towards further work and has suggested some possible solutions – for example, many of the relativization problems may stem from the lack of a clear syntactic category separating relative from non-relative pronouns.

6 Conclusion

In this paper, we provide the first results for probabilistic parsing of modern Portuguese evaluated on significant amounts of human annotated syntactic analyses. We show that an existing probabilistic parser, Bikel's multi-lingual parsing engine, can be readily adapted for Portuguese, and that the accuracy of the parser can be greatly improved with a few relatively straightforward modifications to the parser configuration and to the trees used as training material. Our best configuration on the development material, the AUG-PORT configuration using the tagger tags, achieves 80.6% unlabeled dependency accuracy on unseen test sentences. This result is on par with the accuracy of 80.0% reported for Czech [1].

Much more can be done to improve the parser. In future work, we will perform further modifications to the training trees, such as better handling of discontinuous constituents and introducing finer grained levels of structure instead of the extremely flat trees found in the Floresta. We will also explore lexicalization of models using lemmas as well as full word forms.

Acknowledgements

The authors would like to thank Dan Bikel for advice on modifying and parameterizing the parser. We also thank Susana Afonso, Eckhard Bick, Luis Costa, Diana Santos, and Rui Vilela for their quick and extensive help with the Floresta Sintá(c)tica, and the anonymous reviewers for their feedback.

References

1. Collins, M., Hajic, J., Ramshaw, L., Tillmann, C.: A statistical parser for Czech. In: Proc. of the 37th ACL, College Park, Maryland, USA (1999)
2. Hajic, J.: Building a syntactically annotated corpus: Prague dependency treebank. In: Issues of Valency and Meaning, Karolinum, Prague (1998) 106–132
3. Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proc. of the 35th Annual Meeting of the ACL, Madrid, Spain (1997) 16–23
4. Dubey, A., Keller, F.: Probabilistic parsing for German using sister-head dependencies. In: Proc. of the 41st ACL. (2003) 96–103
5. Dubey, A.: What to do when lexicalization fails: Parsing German with suffix analysis and smoothing. In: Proc. of the 43rd ACL, Ann Arbor, MI (2005) 314–321
6. Arun, A., Keller, F.: Lexicalization in crosslinguistic probabilistic parsing: The case of French. In: Proc. of the 43rd ACL, Ann Arbor, MI, USA (2005) 306–313
7. de Carvalho e Sousa, F.: Analisador sintático estatístico orientado ao núcleo-léxico para a língua portuguesa. Master's thesis, Instituto de Matemática e Estatística da Universidade de São Paulo (2003)
8. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics **29**(4) (2003) 589–638
9. Bonfante, A.G., das Graças Nunes, M.: The implementation process of a statistical parser for Brasilian Portuguese. In: Proc. of the IWPT '01. (2001)
10. Bonfante, A.G.: Parsing Probabilístico para o Português do Brasil. PhD thesis, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (2003)
11. Afonso, S.: Árvores deitadas: Descrição do formato e das opções de análise na Floresta Sintáctica. (2005)
12. Afonso, S., Bick, E., Haber, R., Santos, D.: Floresta sintá(c)tica: A treebank for Portuguese. In Araujo, M.G.R..C.P.S., ed.: Proc. of LREC 2002, Las Palmas de Gran Canaria, Spain (2002) 1698–1703
13. Bikel, D.: Design of a multi-lingual, parallel-processing statistical parsing engine. In: Proc. of the 2nd International Conference on Human Language Technology Research, San Francisco (2002)
14. Bikel, D.: Intricacies of Collins' parsing model. Computational Linguistics **30**(4) (2004) 479–511
15. Bick, E.: The Parsing System PALAVRAS, Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press, Aarhus, Denmark (2000)