Dynamic Propbanking with Deep Linguistic Grammars

António Branco, Sara Silveira, Sérgio Castro, Mariana Avelãs, Clara Pinto and Francisco Costa

University of Lisbon
http://nlx.di.fc.ul.pt

Abstract

In the development of annotated corpora with deep linguistic representations, the category to be assigned to each markable (i.e. the fully fledged grammatical representation of each sentence) is so complex that it cannot be safely constructed manually and the annotation cannot be performed without some supporting application, viz. a computational grammar. This paper discusses and present solutions for the adaptation of deep linguistic grammars as supporting tools in the construction of dynamic propbanks. It reports on the results of an experimental study with a pilot application of this approach to propbanking. Estimated scores for inter-annotator agreement and for development effort are presented.

1 Introduction

Stochastic parsers rely on the availability of annotated corpora both for their training and their evaluation. Such corpora may encompass linguistic information of varying level of complexity, ranging from relatively shallow representation of syntactic constituency (e.g. Penn treebank [12]) to deep representation of fully fledged grammatical analysis that includes advanced semantic representation (e.g. Redwoods HPSG treebank [14]).

The expected trend is that the corpora supporting grammatical analysis will be language resources bearing an increasingly sophisticated linguistic information (Treebanks, PropBanks, LogicalFormBanks, ...). This is in line with the trend observed in natural language processing in general, where annotated corpora that have been developed address issues well beyond the initial part-of-speech level, including issues from the realm of semantics (e.g. corpora annotated with coreference links [11]) and from the realm of discourse (e.g. corpora annotated with dialogue acts tags [7]).

To develop corpora that are annotated with deep linguistic representations, the fully fledged grammatical representation to be assigned to each sentence is so com-

plex and so specific that it cannot be reliably crafted manually piece by piece and the annotation cannot be performed without some supporting application. Such annotation environment has to integrate at least a computational grammar to construct representations that cannot be done by hand, and a user interface to display the parses available and to allow the annotator to selected one of them (e.g. the `[incr tsdb()]` application [13]).

Propbanks are treebanks whose trees have their constituents labeled with semantic role tags. In other words, propbanks are annotated corpora that result from the extension of the annotation associated to the sentences in treebanks by means of an extra set of tags for semantic roles. The seminal work in this area is due to [15] and the semantic roles adopted for arguments were ARG1, ..., ARG4, while the semantic roles used to classify modifiers include tags such as LOC, TMP, MNR, etc.

The construction of propbanks thus offers a new area for the application of deep linguistic grammars to the development of language resources. It also presents specific challenges that need to be addressed.

While word sense disambiguation is a task of semantic categorization for words, propbanking can be seen as a task of semantic categorization for phrases. At their core, tasks of semantic categorization involve semantic ambiguity resolution and deep linguistic grammars are known not to be the most suitable approach to handle semantic categorization tasks. Moreover, any deep linguistic grammar that might be used to support (at least the initial stages of) the construction of a propbank could not rely on stochastic models that would allow it to resolve semantic role assignment, as the training data needed to obtain these models are the materials that are being constructed with the help of the grammar.

An immediate move to ponder would be to extend the grammar so that the different parse trees it produces for a given sentence have their constituents labeled with the different possible semantic roles. The task of propbanking would then be similar to the task of treebanking. The human annotator would go through the parse forest generated and select the appropriate tree, with the appropriate semantic roles in the appropriate constituents. However, this turns out not to be a practically viable option as it would induce an exponential explosion of the size of the parse forest and would severely hurt the efficiency of the grammar.

An alternative to this one-step approach is a two-phase approach. In a first phase, the grammar is used to get at the correct tree for the sentence at stake. Once that representation is selected, in a second phase, its phrases are tagged by the human annotator. This is a scenario that avoids the explosion of the parse forest that would result from the one-step approach. And very importantly, this is a scenario that still allows exploring the contribution of the grammar for propbanking to the maximum extent possible without spurious overgeneration.

The grammar can be used not only to construct the underlying syntactic tree but also to advance the assignment of semantic roles: if its deep linguistic representation is "deep" enough to include the predicate-argument structure of the sentence, the grammar can be used to correctly assign the role labels for its constituents that

are arguments, viz. ARG1 to ARG*n*. By the same token, it can be used to identify the phrases that are modifiers, and hence to automatically select only those constituents that still need to be further manually inspected and specified for their semantic role.

In this paper we report on the adoption of this approach to propbanking and describe the solutions we developed in order to construct a propbank based on a deep linguistic grammar. This involved using an existing deep linguistic grammar and using available tools for the creation of dynamic treebanks, that accommodate the fact that the grammar may evolve and its output may be altered and refined.[1] Accordingly, in this paper we will also describe the solutions developed for the construction of a dynamic propbank that is supported by a grammar that may evolve and be refined along time.

Section 2 reports on the adaptation of a deep linguistic grammar to the task of semi-automatic propbanking and of the tool for dynamic treebanking. In Section 3, we describe the annotation environment developed to assist in the manual completion of a dynamic propbank. Section 4 describes a pilot experimental study of the practical viability of this environment, and Section 5 presents the concluding remarks.

2 Semantic role tagging by the grammar

The deep linguistic grammar used for semi-automatic propbanking was LXGram [5, 6, 4, 3]. This grammar is developed under the grammatical framework of HPSG-Head-Driven Phrase Structure Grammar [16, 17] and uses MRS-Minimal Recursion Semantics [10] for the representation of meaning. Its implementation is undertaken with the grammar development environment LKB-Linguistic Knowledge Builder [9] and the LinGO Grammar Matrix version 0.9 [2] was used as the initial code upon which the grammar is being built.

The evaluation and regression testing of this grammar is done via the application `[incr tsdb()]` [13] that works in tandem with the LKB. Once the sentences are parsed, they are manually disambiguated using this profiling environment, which can then be used to export into text files several views of the syntactic and the semantic analyses obtained by the grammar (e.g. parse trees, feature structures and semantic representations).

As this is an HPSG grammar, there are no explicit categories like Sentence, Noun Phrase, etc. in the grammatical representation. Instead there are feature structures that describe and stand for such categories. The LKB environment, however, has a visualization device for grammatical representations that permits to display tree views where these categories can be made to appear. There are specific configuration files for this device where it can be stated what feature structures should be mapped to what symbols that appear in the nodes of the syntax

[1] We are using the notion of "dynamic" annotation of corpora (for treebanking, propbanking, etc) in the sense worked out for the Redwoods Treebank in [14].

tree displayed. For instance, a constituent with a HEAD feature of the type *verb* and a SUBJ feature with the empty list as its value is categorially a sentence, and it is possible to configure the visualization device for the corresponding tree node to appear with the label S. This is the key facility we explored in this grammar development environment to make semantic role labels appear in the appropriate constituents in the parse tree.

Some of the semantic role labels that are used in PropBank can be obtained from features that describe the semantics of the sentence, namely those used to tag the subject and the complements of predicators, ARG1 to ARGn.

For instance, a verb that is associated to a semantic relation whose first argument is that verb's subject will comply with a constraint like the following:

$$\left[\text{SYNSEM|LOC}\left[\begin{array}{l}\text{CAT|VAL|SUBJ}\left\langle\left[\text{LOC|CONT|HOOK|INDEX}\ \boxed{1}\right]\right\rangle\\ \text{CONT|RELS}\left\{\left[\text{ARG1}\ \boxed{1}\right]\right\}\end{array}\right]\right]$$

This is the underlying piece of information that can be used to assign semantic role labels. Although this piece of information is visible in the feature structures for predicators, it is not visible in the feature structures for the actual phrases that are to be labeled. For this reason, the grammar was expanded with an extra feature (ROLE-LABEL) marking the semantic role label of constituents. In this example, with this sort of verbs, another constraint was added:

$$\left[\text{SYNSEM|LOC}\left[\begin{array}{l}\text{CAT|VAL|SUBJ}\left\langle\left[\text{LOC}\left[\begin{array}{ll}\text{ROLE-LABEL} & \textit{arg1}\\ \text{CONT|HOOK|INDEX} & \boxed{1}\end{array}\right]\right]\right\rangle\\ \text{CONT|RELS}\left\{\left[\text{ARG1}\ \boxed{1}\right]\right\}\end{array}\right]\right]$$

A quite straightforward way to include such semantic role labels in the output tree is by simply adjusting the mapping from feature structures to node labels in the visualization device so that the information contained in the semantic representations is rendered visible in the tree exported.

For the sake of illustration, Figure 1 presents an example of a parse tree exported. All constituents have a category label. The relevant constituents also have a tag describing their syntactic function (SJ for subject, DO for direct object, IO for indirect objects, OBL for oblique complements, PRD for predicative, M for modifier), separated from the category label by a dash. Also separated by a dash, a third tag describes the semantic role that can be obtained from the semantic representations derived by the grammar, ARG1,..., ARG*n* for subcategorized arguments, M for modifiers.

The ARG1,..., ARG*n* tags are similar to the same tags used in PropBank, but note that PropBank starts at ARG0 whereas we start at ARG1. The M is here a portmanteau tag that covers all semantic roles that are possible for modifiers. It corresponds to PropBank tags like LOC, TMP, MNR, etc. As mentioned above, producing these more specific tags by the grammar is not practically viable. Our approach to propbanking consists in manually refining the M tag in a second phase, that is described in the next section.

```
(S (PP-M-M (P (Para))
           (S (NP-SJ-ARG1 (D-SP (a))
                          (N (delegação)))
              (VP-C (V (evitar))
                    (NP-DO-ARG2
                      (D-SP (um))
                      (N' (N (conflito))
                          (AP-M-M (A (armado))))))))))
   (S (PP-M-M (P (em))
              (NP-C (N (Maio))))
      (S (NP-SJ-ARG1 (D-SP (a))
                     (N (ONU)))
         (VP (V' (V' (V (enviou))
                     (ADVP-M-M (ADV (rapidamente))))
                 (NP-DO-ARG2 (N (tropas))))
             (PP-OBL-ARG3
               (P (para))
               (NP-C (D-SP (a))
                     (N (fronteira))))))))
```

Figure 1: Parse tree exported from `[incr tsdb()]` and decorated with semantic role labels. The sentence translates into "For the delegation to_avoid an armed conflict, in May the UN quickly sent troops to the border".

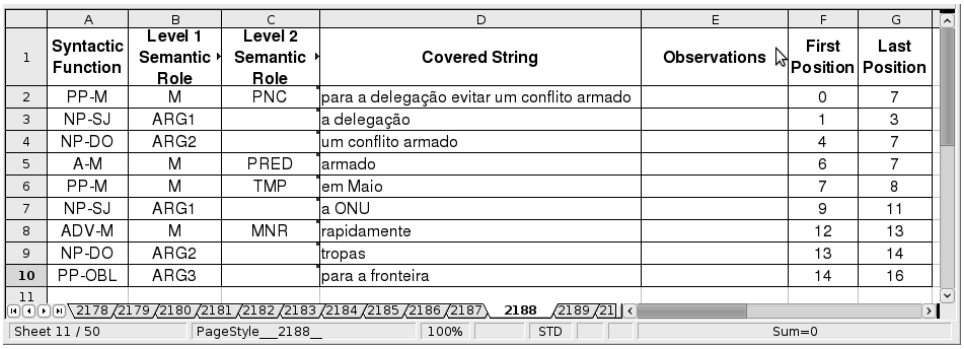
3 Completing semantic role tagging

After the propbanking is advanced in a first phase by the grammar, a completion step follows that consists in the manual specification of the occurrences of the portmanteau tag M in terms of one of the semantic roles available for modifiers in the tagset, LOC, TMP, MNR, etc.

Before proceeding with the descritpion of the second phase of the propbanking, it is worth noting that for the two step annotation methodology we are reporting on, there is nothing essential in the usage of the HPSG framework, the LKB development environment or an MRS semantic representation. Any deep linguistic grammar is suitable to support the first, automatic phase described in the section above provided it delivers deep enough representations, that include at least the semantic roles for arguments.

Turning now to the second phase of the propbanking, this manual annotation is prepared by two tools. A converter from trees into an annotation format compatible with the annotation interface, and a reverser tool for the symmetric operation.

A third tool is also necessary to support the construction of a dynamic Propbank. The development of this annotated corpus is based on a computational grammar which is itself likely to be under development or refinement and to evolve. It should thus be expected that a sentence that received a certain analysis under version n of the grammar may receive a different tree under the subsequent version n+1. This change in grammatical analysis is likely to have an impact on the annotation produced with the support of the grammar in version n and previously stored in the Propbank. Therefore, a third tool is necessary to support this dynamic



	A	B	C	D	E	F	G
1	Syntactic Function	Level 1 Semantic Role	Level 2 Semantic Role	Covered String	Observations	First Position	Last Position
2	PP-M	M	PNC	para a delegação evitar um conflito armado		0	7
3	NP-SJ	ARG1		a delegação		1	3
4	NP-DO	ARG2		um conflito armado		4	7
5	A-M	M	PRED	armado		6	7
6	PP-M	M	TMP	em Maio		7	8
7	NP-SJ	ARG1		a ONU		9	11
8	ADV-M	M	MNR	rapidamente		12	13
9	NP-DO	ARG2		tropas		13	14
10	PP-OBL	ARG3		para a fronteira		14	16
11							

2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 **2188** 2189 21

Sheet 11 / 50 PageStyle__2188__ 100% STD Sum=0

Figure 2: Annotation interface

propbanking.

These tools are described in the next subsections.

3.1 Exporting trees to and from an annotation interface

The annotation interface is based on a basic yet very efficient and powerful enough technology in view of the manual task it is aimed at supporting. A set of sentences to be annotated is presented in a spreadsheet file, with each sentence in a different sheet. The constituents that need to be tagged are displayed in the first column, each constituent in a separate cell. The semantic role labels that were assigned by the grammar are displayed in the second column, aligned with the corresponding constituents. The third column is left blank and its cells in the lines with M in the second column offer a drop down menu from which it is possible to pick the semantic role label with which to specify M.

These spreadsheets are created by the converter tool that takes as input an exported version of the sentences treebanked with `[incr tsdb()]`. For each suite of treebanked sentences, a spreadsheet is created with as many sheets as sentences in that suite. If a given sentence happens not to have received a parse, its sheet only contains its identification number and that sentence. If in turn the sentence received a parse in the treebank, its tree is processed and for each node with a syntactic function that ends in -SJ, -DO, -IO, -M, -OBL, or -PRD, a new line in the sheet is printed.

The sentences that have a parse, i.e. whose propbanking needs to be finalized, are identified by a designated format for the name of the sheets containing them. The annotator only needs to look for those sheets with the name in that format, since only there actual annotation is required to be performed.

As mentioned above, each line has cells automatically filled in, and others to be filled in by the annotator. Each line includes cells with (A) the syntactic category and grammatical function, (B) the semantic role assigned by the grammar, (C) the "level two" of the semantic annotation—the cell to be filled in by the human annotator—, (D) the constituent being tagged, (E) the annotator's observations—

```
(S (PP-M-PNC (P (Para))
             (S (NP-SJ-ARG1 (D-SP (a))
                            (N (delegação)))
                (VP-C (V (evitar))
                      (NP-DO-ARG2
                        (D-SP (um))
                        (N' (N (conflito))
                            (AP-M-PRED (A (armado))))))))))
   (S (PP-M-TMP (P (em))
                (NP-C (N (Maio))))
      (S (NP-SJ-ARG1 (D-SP (a))
                     (N (ONU)))
         (VP (V' (V' (V (enviou))
                     (ADVP-M-MNR (ADV (rapidamente))))
                  (NP-DO-ARG2 (N (tropas))))
             (PP-OBL-ARG3
               (P (para))
               (NP-C (D-SP (a))
                     (N (fronteira)))))))))
```

Figure 3: Parse tree after manual role labeling. Newly introduced tags are -PNC, -PRED, -TMP and -MNR.

free text by the annotator—, and finally (F-G) the begin and the end positions of that phrase in the sentence.

When the propbanking is finalized, the sentences are reverted to the original tree representation, now extended with the newly assigned tags for the semantic roles of modifiers.

This operation is ensured by a reverting tool. This tool parses the data in the sheets of the spreadsheet and, guided by the information in the last column of the sheets on the initial and final position of the phrase, recombines the information stored there with the original information about the parse tree of the sentence. The outcome is a set of restored sentence trees like the one displayed in Figure 1 with the only difference that now level 2 tag M does not occur in them, as in Figure 3. Each one of its occurrences in the original tree generated by the grammar was replaced by the tag with which it was manually specified.

Note that this annotation interface is very flexible and permits to accommodate tagsets other than the one adopted here, which replicates the tagset of the English PropBank in [15]. It permits to specify not only the modifiers with sub-semantic roles LOC, TMP, MNR, etc., but also the arguments with sub-semantic roles like AGENT, PATIENT, EXPERIENCER, etc. One just needs to add extra drop down menus with the required range of tags in the third column cells of the lines already containing the ARG1,..., ARGn tags.

This is thus an annotation interface that can very easily be adapted for arbitrarily complex, multi-layer hierarchical tagsets of semantic role labels, including those that may be extended and get more complex along the development of refined versions of the propbank.

3.2 Supporting the construction of a dynamic propbank

An annotated corpus whose construction is based on a grammar that evolves with time has a dynamic nature. As the grammar gets extended or refined in each new version, the composition of the corpus is likely to evolve as well. For instance, some sentences that got a parse with a previous version n and were annotated with that parse may have no parse in version n+1; sentences that received a parse with version n of the grammar may have that parse tree altered in version n+1; and sentences that had no parse with version n may receive a parse in version n+1. As the grammar evolves, sentences may thus be dropped from the annotated corpus, may have their annotation changed, or new sentences may enter the annotated corpus. Of course, many annotated sentences in the treebank will be kept unchanged in the new versions.

Given the annotation environment adopted for propbanking, in order to minimize the manual effort spent, the two cases that are important to handle in a version n+1 of the propbank (supported by version n+1 of the grammar) are the case of the sentences that have their parse trees changed and the case of the new annotated sentences that entered the treebank. For the remaining propbanked sentences, whose annotation was not altered, they just need to be automatically transfered from version n to version n+1 of the propbank.

To support this dynamic propbanking, a tool was developed that performs the required comparisons between the annotated sentences in version n of the propbank and the outcome of version n+1 of the grammar applied over the pool of the sentence to be propbanked. Moreover, it singles out the sentences that need to receive the attention of the human annotators and transfers the remaining annotated sentences from version n to version n+1.

This comparator tool receives as input the spreadsheets S_n, of the previous version of the propbank, and the spreadsheets S_{n+1}, generated on the basis of version n+1 of the grammar. For each pair of spreadsheets containing the same suite of sentences, every sentence will be checked.

If a sentence maintains its parse tree from version n to version n+1, its propbank annotation will be maintained and transfered from S_n to S_{n+1}. Thus, the annotators do not have to re-annotate these sentences.

If in turn a sentence did not have a parse in version n and received a parse in version n+1, its sheet is signaled in the spreadsheet by means of a designated format for its name. Also, if a sentence receives a parse tree in version n+1 that is different from the parse tree in S_n, its sheet is also signaled. Additionally, for the sake of documentation, the parse tree and its semantic role labels in S_n are copied into S_{n+1} to a position in the corresponding sheet below the new parse tree whose propbanking is to be completed by the annotator.

4 Propbanking with a deep linguistic grammar

The practical viability of the propbanking environment just described was tested in a pilot experimental study. In this study, this environment was used to propbank 807 sentences (5,457 tokens overall; longest sentence with 34 words), extracted from a corpus of newspaper texts with a total of 350 Ktoken and 12 Ksentence [1]. This is an annotated corpus accurately annotated with part-of speech and morphological information and these annotations were used to help constrain the grammar search space.

The propbanking was performed under the annotation methodology consisting in a double blind annotation followed by adjudication. The annotation was produced by two annotators, who hold a degree in the Humanities with formal training in Linguistics and who have more than one year of experience in corpus development and treebanking. The adjudication was done by a third element of the team, who holds a degree in Linguistics and was the main coder of the grammar used.

The 807 sentences (5,457 tokens) were propbanked in 10 hours by the annotators, without any previous period of adaptation to the task in this study. This indicates that a rate of at least 80 sentences (550 tokens) propbanked per hour can be expected, provided these sentences had been previously treebanked in this overall environment for corpus development. A propbank with a typical size of 1 Mtoken could thus be expected to be produced out of a deep linguistic treebank in about 2,000 hours, with an estimated 100 person month of effort for a double blind annotation with adjudication.

This study also allowed getting a first indicative assessment of the level of reliability of the data produced by the annotators, before adjudication, that can be expected for a proposition bank produced under this methodology. The inter-annotator agreement coefficient used was Cohen's κ coefficient [8], calculated by the formula

$$\kappa = \frac{A_o - A_e}{1 - A_e}$$

where A_o is the observed agreement, and A_e is the expected agreement. Expected agreement is the sum of the expected agreement for every tag, where the expected agreement for each tag is the probability of the two annotators to assign that tag by chance. The probability of one annotator assigning a given tag by chance is the proportion of items actually assigned by that annotator with that tag.

The score for the inter-annotator agreement was 0.77 for an observed agreement of 0.83 and an expected agreement of 0.27, calculated over a total number of 334 assigned tags (tokens).

5 Conclusions

Annotated corpora tend to include increasingly sophisticated linguistic information. When developing treebanks with deep linguistic representations, the linguis-

tic information to be associated with each sentence is so complex that it cannot be safely done manually and the annotation has to rely on some computational grammar supporting it. In this paper we discussed how to extend the utilization of deep linguistic grammars as supporting tools also for the annotation of propbanks.

In order to explore the contribution of the grammar for propbanking to the maximum extent possible without hurting its efficiency, the propbanking environment studied was based on two phases. In a first phase, the grammar is used to treebank the sentence. As the grammatical representations produced by the grammar already include predicate-argument structures, this information was explored to annotate the argument phrases with the semantic role labels ARG1,...,ARG*n* for arguments. In a second phase, the parse trees were converted to an interface format that singled out the modifier phrases, whose propbanking still needs to be completed by the human annotators.

As the grammar is typically an application that will be extended and refined over time, the propbanking environment needs to be prepared to support the construction of a dynamic propbank. This is achieved with the help of a comparison tool that permits drawing the attention of the annotators only for the sentences newly entered in the treebank.

This propbanking environment was tested in a preliminary experimental study. In this study, though the annotators and the adjudicator were performing their tasks for the very first time, without previous substantial training that could be gathered after continued work, the results of this pilot study were very promising. They suggest that, provided human annotators have been sufficiently exercised over a large enough portion of corpus, this environment will support the propbanking of over 80 sentences per hour and permits to expect a reliable level of inter-annotator agreement, that will rise over 0.77 in terms of the κ coefficent. A corpus with a large size of 1 Mtoken could thus be expected to be produced out of a deep linguistic treebank with less than 100 person month of annotation effort with a double blind annotation methodology.

References

[1] Florbela Barreto, António Branco, Eduardo Ferreira, Amália Mendes, Maria Fernanda Nascimento, Filipe Nunes, and João Silva. Open resources and tools for the shallow processing of portuguese: the TagShare project. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.

[2] Emily M. Bender, Dan Flickinger, and Stephan Oepen. The Grammar Matrix: An open-source starter-kit for the development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

[3] António Branco and Francisco Costa. Accommodating language variation in deep processing. In Tracy Holloway King and Emily M. Bender, editors, *Proceedings of the Grammar Engineering Across Frameworks Workshop (GEAF07)*, pages 67–86, Stanford, 2007. CSLI Publications.

[4] António Branco and Francisco Costa. Self- or pre-tuning? deep linguistic processing of language variants. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 57–64, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[5] António Branco and Francisco Costa. A computational grammar for deep linguistic processing of portuguese: LXGram. In *Technical Reports Series*. University of Lisbon, Department of Informatics, 2008.

[6] António Branco and Francisco Costa. LXGram in the shared task "comparing semantic representations" of STEP2008. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing*, pages 299–314, London, 2008. College Publications.

[7] Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–32, 1997.

[8] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[9] Ann Copestake. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, 2002.

[10] Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. Minimal Recursion Semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332, 2005.

[11] Lynette Hirschman, Patricia Robinson, John Burger, and Marc Vilain. Automating coreference: The role of annotated training data. In *Proceedings of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 1997.

[12] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[13] Stephan Oepen and John Carroll. Parser engineering and performance profiling. *Natural Language Engineering*, 6(1):81–98, 2000. (Special Issue on Efficient Processing with HPSG).

[14] Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. The LinGO Redwoods treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 1253–7, Taipei, Taiwan, 2002.

[15] Martha Palmer, Dan Gildea, and Paul Kingsbury. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1), 2005.

[16] Carl Pollard and Ivan Sag. *Information-Based Syntax and Semantics, Vol. 1*. CSLI Publications, Stanford, 1987.

[17] Carl Pollard and Ivan Sag. *Head-Driven Phrase Structure Grammar*. Chicago University Press and CSLI Publications, Stanford, 1994.