Detecting Errors in Semantic Annotation

Markus Dickinson, Chong Min Lee

Indiana University, Georgetown University
md7@indiana.edu, cml54@georgetown.edu

Abstract

We develop a method for detecting errors in semantic predicate-argument annotation, based on the variation n-gram error detection method. After establishing an appropriate data representation, we detect inconsistencies by searching for identical text with varying annotation. By remaining data-driven, we are able to detect inconsistencies arising from errors at lower layers of annotation.

1. Introduction and Motivation

Corpora with semantic annotation (e.g., Baker et al., 1998; Palmer et al., 2005; Burchardt et al., 2006; Taulé et al., 2005) are becoming increasingly relevant in natural language processing. Semantic role labeling—used for tasks such as information extraction (e.g., Surdeanu et al., 2003), machine translation (e.g., Komachi et al., 2006), and question answering (e.g., Narayanan and Harabagiu, 2004)—requires corpora annotated with predicate-argument structure for training and testing data (see, e.g., Carreras and Màrquez, 2005; Carreras and Màrquez, 2004; Gildea and Jurafsky, 2002; Xue and Palmer, 2004; Toutanova et al., 2005; Pradhan et al., 2005). Not only, then, are there clear applications for predicate-argument structure, but as semantically-annotated corpora are generally built on top of syntactic annotation, such corpora have enormous potential as sources of linguistic data for theoretical research.

However, annotating semantic corpora is non-trivial. Given the difficulty of determining certain predicate-argument relations, e.g., modifier (ArgM) predications in the Proposition Bank (PropBank, Palmer et al., 2005), and of selecting an underlying theory (see, e.g., Burchardt et al., 2006), there is a need to support annotation endeavors by providing feedback on the annotation schemes. Relatedly, in the process of annotating the corpus, annotation may have been inconsistently applied, leading to errors. These errors can affect the annotation's uses: for other annotation types, errors can lower the performance of applications training on the corpora (e.g., van Halteren et al., 2001; Dickinson and Meurers, 2005b), prompting workarounds to deal with noisy data (e.g., Hogan, 2007; Banko and Moore, 2004). Moreover, we cannot accurately evaluate technology using a corpus with a significant number of errors as the "gold standard" (Padro and Marquez, 1998; Květǒn and Oliva, 2002). Furthermore, given that semantic annotation has generally been built on top of POS and syntactic layers of annotation, it is important to see how the layers interact, especially in terms of errors in one layer affecting another. Thus, there is a need to investigate the quality of annotation in semantically-annotated corpora.

While previous research has detected errors in part-of-speech and syntactically-annotated corpora (see Dickinson, 2005, ch. 1), to our knowledge there has been no work on automatically detecting errors in semantically-annotated corpora in a corpus-independent way. Yet this is vitally important. On the one hand, as mentioned, it is important to detect errors, as they are detrimental to semantic role labeling tasks, and the process of error detection can provide feedback on the annotation scheme. On the other, investigating error detection for semantic annotation can provide insights into the use of this annotation. Additionally, it is important to know whether error detection methods will scale up to semantic annotation, as syntactic and semantic annotation seem to rely on different information. Thus, we develop an error detection method for semantic annotation, basing it on the variation n-gram method (Dickinson and Meurers, 2005a). In section 2., we outline this method, which detects items occurring multiple times in the corpus with varying annotation, and in section 3. we discuss how to extend it to semantic annotation. Section 4. provides results showing the successfulness of this method.

2. Background

The variation n-gram method (Dickinson and Meurers, 2003a,b, 2005a) detects items occurring multiple times in the corpus with varying annotation, and these are the so-called *variation nuclei*. A nucleus with its repeated surrounding context is a *variation n-gram*. Variations in annotation are classified as errors or genuine ambiguities using a basic heuristic requiring at least one word of context on each side of the nucleus.

Following Dickinson and Meurers (2003b), we illustrate the variation n-gram approach using the Wall Street Journal in the Penn Treebank (Taylor et al., 2003): the variation nucleus *next Tuesday* in the variation trigram *maturity next Tuesday* has a labeling error, being twice labeled NP and once PP. A bracketing error occurs for *jolt last month from*, as shown in Figure 1: the variation nucleus *last month* occurs twice in this local context, once with the label NP and once as a non-constituent, which we represent with the special label NIL.

The basic heuristic used to classify cases as errors or non-errors requires one word of context around every word in the nucleus; this is referred to as the *non-fringe* heuristic, as no nucleus words are on the fringe of the n-gram. In Figure 1, for example, the nucleus *last month* is surrounded by *jolt* and *from* in both corpus instances, indicating an error. Although Dickinson and Meurers (2003a) expand the context as far as possible ("trust long contexts"), Dickinson (2005) shows that using only local context results in

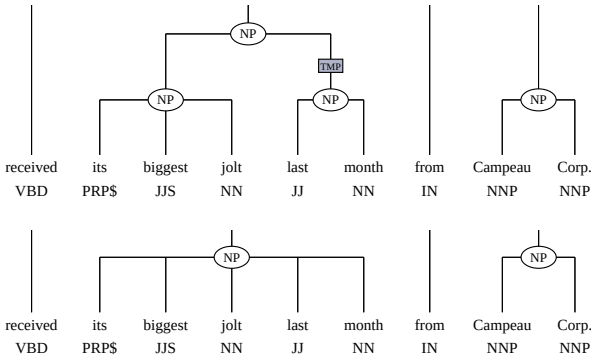

Figure 1: Bracketing error detected in the WSJ

nearly equally high precision.[1] This local context heuristic receives support from research on language acquisition showing that such context frames are used by infants during lexical category learning (Mintz, 2003, 2002).

3. Detecting semantic annotation errors

Being data-driven, this method relies on a single mapping between text and annotation, but semantic annotation is made of non-uniform components. For each verb in PropBank, for example, the annotation contains: 1) the verb sense, 2) the span of each argument, and 3) argument label names. Identifying a verb sense, however, is a completely different task from identifying an argument. Semantic role labeling tasks have split predicate-argument annotation from verb sense tagging (e.g., Morante and van den Bosch, 2007), and we follow that split. As the relations between the verb and its arguments are generally determined by local context, the method is most readily applicable to argument identification and labeling, and so we focus on that. Extending the work to verb sense inconsistency detection is left for future work.

Consider example (1a) from PropBank: *lending practices* functions as the (non-agentive) subject of *vary* (Arg1); *widely* indicates the extent of the variation (Arg2-EXT); and *by location* is a modifier (ArgM) specifying the manner (MNR) of the action. The verb *vary* is annotated as having sense *vary.01*, but we ignore the sense and focus on the bracketing and labeling.[2] As can be seen, the representation is thus closely related to both constituency and dependency annotation.

(1) a. [$_{Arg1}$ lending practices] **vary**/vary.01 [$_{Arg2-EXT}$ widely] [$_{ArgM-MNR}$ by location]

b. [$_{Arg1}$ lending practices] **vary**/vary.01 [$_{ArgM-MNR}$ widely] [$_{ArgM-MNR}$ by location]

Given the different labeling of *widely* in (1b), we need some way to systematically find such variation.

[1]The precision is estimated in Dickinson (2005) as 92.8%, whereas Dickinson and Meurers (2003a) report 97.6% precision, but a third as many cases, for longer contexts.

[2]For the remainder of the examples, we will therefore leave out the verb sense.

3.1. Argument labeling variation

Although the corpus is indexed by each verb with all its arguments, we can view the annotation as consisting of multiple pairwise relations between a verb and a single argument. While the various arguments are not completely independent, they often have no bearing on each other. The manner adverbial *by location* in (1), for example, does not affect the annotation of *lending practices*. Thus, to adapt the variation n-gram method for predicate-argument annotation, we define a nucleus as consisting of both a verb and a single argument. The nuclei for this sentence, then, are: *lending practices vary*, *vary widely*, and *vary by location*. With this definition, semantic annotation involves potentially discontinuous elements (e.g., *vary by location*), prompting us to use the variation n-gram algorithm as developed for discontinuous syntactic constituency annotation (Dickinson and Meurers, 2005a).

To use this method, we have to map the nucleus to a label, and we can start by assigning each nucleus the argument label, e.g., Arg0, but this is not sufficient. A semantic "constituent" contains non-uniform elements, unlike a syntactic constituent, which—when no head is marked—is a single string where no member is more prominent than another. In principle, for semantic annotation, two identical strings could both have some label, but identify the argument differently. Thus, we include the position of the verb in the nucleus; e.g., the label of the nucleus *vary widely* in (1a) is ArgM-MNR-0. Actually, we put every verb position into the label, as both verbs and arguments can contain multiple, discontinuous elements: the verb can wrap around the argument (*pick it up*) or the argument can wrap around the verb (see Babko-Malaya, 2005, p. 20). Although variation in identification of the verb is unlikely, this data representation ensures completeness of the method.

3.2. Argument identification variation

While using this representation for the variation n-gram method will find errors in labeling (e.g., Arg0 vs. Arg1), we also want to find errors where an argument has gone unidentified, or covers a different stretch of comparable text in another part of the corpus. Following Dickinson and Meurers (2005a), we represent a string not labeled as an argument by assigning it the label NIL. We are only interested in NIL strings which are identical to previously-identified verb-argument pairs, and so we use the techniques in Dickinson and Meurers (2005a) to efficiently find such NIL strings.

We can see an example in (2): in (2a), *net income in its first half rose* is annotated as Arg1-6, whereas in (2b) it is not completely annotated as an argument and thus receives the label NIL.[3] This allows us to spot inconsistency in the corpus annotation. As we will see in section 4., many of these variations in NIL strings are rooted in errors in lower layers of annotation.

(2) a. [$_{Arg1}$ net income in its first half] **rose** 59 %

b. [$_{Arg1}$ net income] in its first half **rose** 8.9 %

[3]We underline the variation nucleus and boldface the verb.

Finally, because we want to cover every verb and its potential arguments, we have to deal with cases where the verb contains more than one element, i.e., variation in phrasal verb (PV) identification. Although not annotated as such, the particle is implicitly an argument of the verb, and we thus created a special relation to capture these cases. For example, the particle verb combination *get back* is re-coded as a PV relation between *get* and *back*, as in (3a), and we can thus find instances of *get back* annotated differently, as in (3b).

(3) a. we could **get** [$_{PV}$ back] to investing
b. an excuse for people to **get** [$_{ArgM-ADV}$ back] to reality

3.3. Heuristics for disambiguating strings

The method as outlined so far generates far too many false positives, especially given the great number of discontinuous NIL strings which are clearly not comparable. To narrow our search for comparable strings, we need some contextual information. As a simple way to identify errors, we can require one word of identical context to surround every word in the variation nucleus in order to flag it as a potential error (Dickinson and Meurers, 2005a). This "shortest non-fringe" heuristic, however, is rather strict.

Thus, in order to increase recall, we experiment with another heuristic. Since we are dealing with errors in the relation between the verb and the argument, we have to ask what is the most crucial information. Recall that there are really two potential ways that something can be erroneous: an error in the labeling (or non-labeling) of the argument, or an error in the identification of the argument. The identification of the verb is not really a major issue, given that verbs are generally only a single word and that they drive the analysis. Thus, the context surrounding the argument is more important. Indeed, once we know the verb and the (supposed) argument, we can wonder whether any context around the verb is necessary; knowing that it is potentially related to the argument may be sufficient.

In (4), for example, the nucleus *substantially reduce* does not depend on what follows the verb; the relation between the verb and the argument is context enough.

(4) a. That could [$_{Arg2-MNR}$ substantially] **reduce** the value of the television assets .
b. the proposed acquisition could [$_{ArgM-MNR}$ substantially] **reduce** competition ...

Likewise, what kind of court ruling in (5) is at issue is not important to determining whether *court* is an argument of *ruling* (5a) or not (5b).

(5) a. from a [$_{Arg0}$ court] **ruling** on a tax dispute
b. stems from a court **ruling** that *T* found ...

As for argument context, consider (6)[4]: in (6a), *officials* has no modifier, but in (6b) it does, making it a different argument. Thus, we need identical context around the argument to avoid false positives in argument identification.

(6) a. Finnair would receive SAS shares valued * at the same amount , [$_{Arg0}$ officials] **said** 0 *T* .
b. ... [$_{Arg0}$ government officials] **said** 0 they had n't noticed any surge in the filings .

Thus, to increase recall, we require one word of context around the argument, but not the verb. We refer to this as the *argument context heuristic*.

One final point is in order about the argument context heuristic. It is possible to find variation in identifying the verbal head of the semantic relationship, in which case it is not prima facie clear where to add additional context since we are only adding context around the argument. Our solution is to use the argument context heuristic for either potential arguments, and identify the one which still results in variation. For example, in (7), both *continued* and *strengthening* can be the verbal head, and so we search for context around either potential head. In this case, both have identical surrounding context, so the argument context heuristic identifies nothing more than the shortest non-fringe heuristic.

(7) a. the dollar 's [$_{ArgM-MNR}$ continued] **strengthening** reduced world-wide sales growth by three percentage points
b. the dollar 's **continued** [$_{Arg1}$ strengthening] reduced world-wide sales growth by three percentage points

In practice, this situation almost never occurs—in fact, this example is the only case out of 947 variations (see section 4.2.) where there is variation in identifying the verb. This example, however, points out one potentially useful quirk about our method—this variation occurs within the same sentence, i.e., for the same tokens. One might consider filtering these cases out from the method, but given the traditional assumption that only one element may be the head of a dependency, flagging these cases can highlight non-traditional aspects of the annotation scheme, and so we do nothing to remove them.

While we would like to abstract the nucleus to even more general properties, e.g., POS categories (cf. Boyd et al., 2007), given that the labeling is highly verb-dependent, we cannot abstract the verb to anything other than a lemma. That is, distributional categories even seem to be too coarse, although there might be more semantically meaningful abstractions to consider.

4. Evaluation

At least two major corpora with semantic predicate-argument annotation are currently available for English: FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). Both label verbs and their arguments, but one big difference between the corpora is the granularity, or specificity, of their semantic roles. FrameNet gives a specific name to each semantic role, such as *Experiencer* or *Sender*. PropBank, on the other hand, uses more generic labeling for its arguments, such as Arg0 and Arg1, as highly specific argument labels make semantic role labeling more difficult. Following research on semantic role labeling using PropBank as a data source (e.g., Pradhan et al., 2005; Toutanova

b.

[4] *T*, *, and *0* refer to empty elements inserted by annotators.

et al., 2005; Xue and Palmer, 2004), we use PropBank in our error detection work.

4.1. Annotation-specific issues

Before we turn to the results, we need to discuss the format of the annotation for our case study. First is the issue of function labels, or secondary predications, in PropBank. As described above, these labels indicate additional information about the argument's relation to the verb. For error detection for syntactic annotation on the Penn Treebank (Dickinson and Meurers, 2003b), such function labels are removed from consideration. In that context, the removal of function labels was motivated by the fact that the variation nucleus contained no information about the word the constituent was dependent upon. We, however, retain function labels, or secondary predications, in the PropBank labels: our variation nucleus consists of a verb and its argument, giving the information we need to make a decision about the function.

Secondly, PropBank uses null elements to encode non-local relations such as traces of movement and right node raising, as well as other complex linguistic phenomena. We opt for the simplest treatment here, filtering out any nuclei with null elements, to avoid positing variation between truly different arguments since null elements give no indication of the semantics (cf. Dickinson and Meurers, 2003b). In the future, one could consider working through a chain of traces to get to the real words serving as a semantic argument, in cases where the anaphora relations are clearly laid out. Even if this is done, however, it is not clear how one is to treat the context, and thus we leave it for future work.

4.2. Results and insights

After removing null element nuclei, we find 43,825 variation nuclei, an enormous number of variations, most of which are acceptable, as the contexts are truly different. Requiring context surrounding the whole nucleus (i.e., the shortest non-fringe heuristic) results in 369 shortest non-fringe variation nuclei, but this reduces the variation set too much. While the precision is likely high, the recall can be improved.

The expanded heuristic of only requiring a single word of context around the argument (the argument context heuristic) turns up 947 variation nuclei, increasing recall nearly threefold. Of the 947 variation nuclei, 835 cases involve argument identification variation, i.e., NIL labels, and only 127 feature variation between labels;[5] we discuss below the reasons for a higher number of identification inconsistencies. From this set of 947 variations, we sampled 100 cases, and we find that 69% point to errors, or inconsistencies. This heuristic thus successfully increases error detection recall, using only very simple pieces of information. The local context of the argument seems to be generally sufficient context, given that the verb is already known, and this expansion could thus potentially also be used for dependency annotation.

[5]This adds up to 962 because 15 variations have variation both in identification and in labeling.

Discussion The method turns up many cases which raise issues for the annotation scheme, e.g., the determination of modifiers (ArgM), as in example (4) above, where *substantially* modifies *reduce*, but it is not clear exactly how.

An overwhelming number of inconsistencies, however, arise from lower-layer annotation errors propagating to the PropBank layer, leading to erroneous NIL strings (i.e., argument identification inconsistencies). Only verbs are annotated in PropBank, but many words were inconsistently POS-annotated. For example, the phrase in (8) varies in the tagging of *coming*, between JJ (adjective) and VBG (*-ing* verb), and every JJ case is left unmarked in PropBank, as in (8b), whereas the VBG cases have an argument, as in (8a). In fact, of the 69 inconsistent cases, a full 29 of them (42%) are due to erroneous POS annotation.

(8) a. **coming**/VBG [$_{Arg1}$ months] ,
b. coming/JJ months ,

This happens at the syntactic layer, too, as 13 of the 69 inconsistencies (19%) are due to syntactic errors that have propagated to the semantic level. This is shown in (9), where a difference in PP identification leads to a difference in argument identification.

(9) a. The following ... are tentatively **scheduled** * [$_{Arg2-for}$ [$_{PP}$ for sale]] this week
b. The following ... are tentatively **scheduled** * [$_{Arg2-for}$ [$_{PP}$ for [$_{NP}$ sale this week]]]

From this, we can see the dangers of building annotation on top of erroneous annotation: putting POS and syntactic annotation together, 61% of the inconsistencies (42/69) stem from erroneous annotation at lower levels. To detect these problematic cases, it is important to maintain a mapping directly from the data to the semantic annotation, i.e., not rely on syntactic or POS annotation. Being data-driven in this way thus nicely complements the techniques in Babko-Malaya et al. (2006), which rely on detecting inconsistencies between syntactic and semantic annotation (cf. also Dickinson and Meurers, 2005b), since our inconsistencies feature agreement between the layers of annotation.

The method has its limitations, however, which we discuss here. First, some verbs are ambiguous in whether they take arguments and what type of arguments they take. For instance, *have* can be a main verb, as in (10a), which takes arguments, or an auxiliary, which does not, as in (10b), where the main verb *expected* takes the subject argument.

(10) a. [$_{Arg1}$ Analysts] **had** mixed responses to the results .
b. [$_{Arg1}$ Analysts] had **expected** Consolidated to post a slim profit ...

Secondly, many cases of argument identification ambiguity are rooted in difficulties resolving syntactic ambiguity; what the argument is composed of depends upon whether the following phrase attaches with it or not. In example (11), for instance, whether *a buyer* is a complete argument of *seeking* depends upon how the following *for* phrase attaches: in (11a) it attaches to the verb, and in (11b) it attaches to *buyer*, resulting in different syntactic and semantic arguments. One could rely upon the syntactic annotation

to rule out such false positives, but as we have already seen, there are many syntactic errors.

(11) a. which had been **seeking** [$_{Arg1}$ a buyer] [$_{PP}$ for several months]

b. **seeking** [$_{Arg1}$ a buyer for only its shares]

Finally, some argument relations depend upon the sense of the verb, which in turn depends upon the other arguments the verb has. We can see this illustrated in example (12), where two different senses of *return* result in different labeling of *he*, despite the very coarse label set. For languages with verbs which change their subcategorization frames and annotation schemes which match the shifts, error detection may likely overflag more cases.

(12) a. [$_{Arg0}$ he] will **return** Kidder to prominence

b. [$_{Arg1}$ he] will **return** to his old bench

Although it would decrease recall, the shortest non-fringe heuristic could successfully rule out many of these false positives, as it would in (12). A complete solution to these problems, however, requires significantly more technology than we employ. Yet, without such a solution, we still find a significant number of errors with good precision.

5. Summary and Outlook

We have investigated an approach to error detection for semantic annotation, which relies on identical text varying in annotation, and we have found the method to be quite successful. While the non-parallel nature of semantic annotation presented challenges, once we were able to define units of data for comparison, it was relatively straightforward to search for variation. Using a more relaxed heuristic resulted in a clear gain in error detection recall. We discovered in the process that many inconsistencies stemmed from erroneous annotation at lower layers.

To validate the method further, one would want to test it on additional corpora. As mentioned above, a corpus such as FrameNet would be ideal, in order to determine the effect of more fine-grained distinctions in the set of argument labels.

Another avenue of future research would be to increase recall, perhaps by using distributional categories instead of words (Boyd et al., 2007). However, the nature of semantic annotation is such that POS tags are less informative for it than for syntactic annotation.

More informative are the actual words involved in the semantic relations. In fact, it seems that only the heads of arguments are the relevant items for determining what the label is. Thus, to increase recall further for argument labeling error detection and to sidestep issues of ambiguous argument identification (as in (11)), one can consider using only the heads of arguments to see if they are consistently annotated. For example (11), for instance, the nucleus would be *seeking buyer*, highlighting that both cases should be Arg1, and bypassing issues of how the following PP should attach. Such an extension would also likely be more useful for identifying variation in sense annotation, as one could consider more arguments together.

References

Babko-Malaya, Olga (2005). *PropBank Annotation Guidelines*. Tech. rep.

Babko-Malaya, Olga, Ann Bies, Ann Taylor, Szuting Yi, Martha Palmer, Mitch Marcus, Seth Kulick and Libin Shen (2006). Issues in Synchronizing the English Treebank and PropBank. In *Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006*. Sydney, pp. 70–77.

Baker, Collin F., Charles J. Fillmore and John B. Lowe (1998). The Berkeley FrameNet Project. In *Proceedings of ACL-98*. Montreal, pp. 86–90.

Banko, Michele and Robert C. Moore (2004). Part-of-Speech Tagging in Context. In *Proceedings of COLING 2004*. Geneva, Switzerland, pp. 556–561.

Boyd, Adriane, Markus Dickinson and Detmar Meurers (2007). Increasing the Recall of Corpus Annotation Error Detection. In *Proceedings of the Sixth Workshop on Treebanks and Linguistic Theories (TLT 2007)*. Bergen, Norway, pp. 19–30.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Pado and Manfred Pinkal (2006). The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of LREC-06*. Genoa.

Carreras, Xavier and Lluís Màrquez (2004). Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In Hwee Tou Ng and Ellen Riloff (eds.), *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*. Boston, Massachusetts, USA: Association for Computational Linguistics, pp. 89–97.

Carreras, Xavier and Lluís Màrquez (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*. Ann Arbor, Michigan, pp. 152–164.

Dickinson, Markus (2005). Error detection and correction in annotated corpora. Ph.D. thesis, The Ohio State University.

Dickinson, Markus and W. Detmar Meurers (2003a). Detecting Errors in Part-of-Speech Annotation. In *Proceedings of EACL-03*. Budapest, pp. 107–114.

Dickinson, Markus and W. Detmar Meurers (2003b). Detecting Inconsistencies in Treebanks. In *Proceedings of TLT-03*. Växjö, Sweden, pp. 45–56.

Dickinson, Markus and W. Detmar Meurers (2005a). Detecting Errors in Discontinuous Structural Annotation. In *Proceedings of ACL-05*. pp. 322–329.

Dickinson, Markus and W. Detmar Meurers (2005b). Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters. In *Proceedings of TLT-05*. Barcelona.

Gildea, Daniel and Daniel Jurafsky (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics* 28(4), 245–288.

Hogan, Deirdre (2007). Coordinate Noun Phrase Disambiguation in a Generative Parsing Model. In *Proceedings of ACL-07*. Prague, pp. 680–687.

Komachi, Mamoru, Masaaki Nagata and Yuji Matsumoto (2006). Phrase Reordering for Statisitcal Machine Translation Based on Predicate-Argument Structure. In *Proceedings of the International Workshop on Spoken Language Translation*. Kyoto, Japan, pp. 77–82.

Květǒn, Pavel and Karel Oliva (2002). Achieving an Almost Correct PoS-Tagged Corpus. In Petr Sojka, Ivan Kopeček and Karel Pala (eds.), *Text, Speech and Dialogue (TSD)*. Heidelberg: Springer, no. 2448 in Lecture Notes in Artificial Intelligence (LNAI), pp. 19–26.

Mintz, Toben H. (2002). Category induction from distributional cues in an artificial language. *Memory & Cognition* 30, 678–686.

Mintz, Toben H. (2003). Frequent frames as a cue for grammatical categories in child directed speech. *Cognition* 90, 91–117.

Morante, Roser and Antal van den Bosch (2007). Memory-Based Semantic Role Labeling of Catalan and Spanish. In *Proceedings of RANLP-07*. pp. 388–394.

Narayanan, Srini and Sanda Harabagiu (2004). Question Answering based on Semantic Structures. In *International Conference on Computational Linguistics (COLING 2004)*. Geneva, Switzerland.

Padro, Lluis and Lluis Marquez (1998). On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In *COLING/ACL-98*.

Palmer, Martha, Daniel Gildea and Paul Kingsbury (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31(1), 71–105.

Pradhan, Sameer, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin and Daniel Jurafsky (2005). Support Vector Learning for Semantic Argument Classification. *Machine Learning* 60(1), 11–39.

Surdeanu, Mihai, Sanda Harabagiu, John Williams and Paul Aarseth (2003). Using Predicate-Argument Structures for Information Extraction. In *Proceedings of ACL-03*.

Taulé, M., J. Aparicio, J. Castellví and M.A. Martí (2005). Mapping syntactic functions into semantic roles. In *Proceedings of TLT-05*. Barcelona.

Taylor, Ann, Mitchell Marcus and Beatrice Santorini (2003). The Penn Treebank: An Overview. In Anne Abeillé (ed.), *Treebanks: Building and using syntactically annotated corpora*, Kluwer, pp. 5–22.

Toutanova, Kristina, Aria Haghighi and Christopher Manning (2005). Joint Learning Improves Semantic Role Labeling. In *Proceedings of ACL-05*. Ann Arbor, Michigan, pp. 589–596.

van Halteren, Hans, Walter Daelemans and Jakub Zavrel (2001). Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics* 27(2), 199–229.

Xue, Nianwen and Martha Palmer (2004). Calibrating Features for Semantic Role Labeling. In Dekang Lin and Dekai Wu (eds.), *Proceedings of EMNLP 2004*. Barcelona, pp. 88–94.