Unsupervised Learning of Verb Argument Structures

Thiago Alexandre Salgueiro Pardo[1], Daniel Marcu[2], Maria das Graças Volpe Nunes[1]

[1] Núcleo Interinstitucional de Lingüística Computacional (NILC)
CP 668 – ICMC-USP, 13.560-970 São Carlos, SP, Brasil
http://www.nilc.icmc.usp.br

[2] Information Sciences Institute (ISI)
4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292
http://www.isi.edu

taspardo@gmail.com, marcu@isi.edu, gracan@icmc.usp.br

Abstract. We present a statistical generative model for unsupervised learning of verb argument structures. The model was used to automatically induce the argument structures for the 1,500 most frequent verbs of English. In an evaluation carried out for a representative sample of verbs, more than 90% of the induced argument structures were judged correct by human subjects. The induced structures also overlap significantly with those in PropBank, exhibiting some correct patterns of usage that are not present in this manually developed semantic resource.

1 Introduction

Inspired by the impact that the availability of Penn Treebank (Marcus et al., 1993; Marcus, 1994) had on syntactic parsing, several efforts have recently focused on the creation of semantically annotated resources. The annotation of verb arguments, their roles, and preferential linguistic behaviors represents a significant fraction of these efforts. The annotations that we are focusing on here pertain to the argument structures of a verb. In particular, we look for the words/concepts that constitute the arguments required by the verbs when these are used in real sentences.

The determination of verb argument structures has been shown to be a hard task for several reasons. Little agreement exists with respect to (a) how many canonical usages a verb has, (b) which arguments are really required by a verb and (c) in what order they may be realized in sentences. For instance, examples (1)-(3) show some patterns of usage for the verb *bought*.

(1) He had bought them gifts.
(2) He bought it 40 years ago.
(3) About 8 million home water heaters are bought each year.

Intuitively, one can induce from these examples that the object/thing that is bought ("gifts" in sentence (1), "it" in sentence (2), and "about 8 million home water heaters" in sentence (3)) is more likely to be a required argument for the verb than the time when the buying event occurred, since the thing bought is specified in all the cases and time is not. The examples also show the variation in the order in which the arguments are realized: in (1) and (2), the thing bought is stated after the verb; in (3), it is stated before the verb. Ideally, all the possibilities should be acknowledged in the semantic specification of verbs.

There is also little agreement with respect to how the arguments should be labeled. Figures 1, 2, and 3 show the information associated with the verb “buy” in FrameNet (Baker et al., 1998), VerbNet (Kipper et al., 2000), and PropBank (Kingsbury and Palmer, 2002), respectively. These are large scale projects that aim at developing semantic information repositories for verbs, mainly. FrameNet shows the pattern in which a verb occurs and provides representative examples; the resource also organizes the verbs into a hierarchy that implicitly encodes how verb structures can be inherited from ancestors. VerbNet shows the thematic roles the verb asks for, their semantic features, and possible subcategorization frames; VerbNet also provides examples for each categorization frame. PropBank makes explicit the argument roles of a verb, the possible subcategorization frames, and provides examples for each one. PropBank also distinguishes between obligatory verb arguments and optional ones, i.e., adjuncts. The adjunct in Figure 3, for example, is the ArgM-MNR argument (i.e., argument of manner). By inspecting Figures 1-3, it is not difficult to see that little agreement exist with respect to the ontological status of argument labels. What is ARG1 after all? *Goods*? A *Theme*? Or the *Thing Bought*? What is the most appropriate level of abstraction for argument labels?

Typical pattern:
BUYER buys GOODS from SELLER for MONEY

Example:
Abby bought a car from Robin for $5,000.

Figure 1. FrameNet annotation for the verb *buy*

Thematic Roles:
Agent[+animate OR organization],
Asset[-location -region],
Beneficiary[+animate OR +organization],
Source[+concrete], Theme[]

Frames:
Basic Transitive:
"Carmen bought a dress" (Agent, Theme)

Figure 2. VerbNet annotation for the verb *buy*

Roles:
Arg0:buyer **Arg1**:thing bought **Arg2**:seller
Arg3:price paid **Arg4**:benefactive

Examples:
Intransitive:
Consumers who buy at this level are more educated than they were.

Arg0:	Consumers
REL:	buy
ArgM-MNR:	at this level

Figure 3. PropBank annotation for the verb *buy*

Given the difficulty of the task, it is not surprising that FrameNet, VerbNet, and PropBank have been manually built. However, some research efforts have targeted the problem of automatic (Brent, 1991; Resnik, 1992; Grishman and Sterling, 1992; Manning, 1993; Framis, 1994; Briscoe and Carroll, 1997; Rooth et al., 1999; McCarthy, 2000; Sarkar and Zeman, 2000; Merlo and Stevenson, 2001; Sarkar and Tripasai, 2002; Gildea, 2002) and semi-automatic (Korhonen, 2002; Green et al., 2004; Gomez, 2004) verb argument structures induction (including the related task of verb subcategorization frames learning). In general, these approaches rely on syntactic information and/or subcategorization dictionaries for identifying the arguments of a verb in a sentence, and/or assume as known the structure types in terms of number and order of arguments a verb can assume. The main goal in these approaches is to identify the lexemes that are most likely to fill a given verb argument slot. Some researchers (Grishman and Sterling, 1994; Framis, 1994; Lapata, 1999; Gomez, 2004) try to go beyond these lexemes and generalize the structures that are learned, by computing the similarity between the words occurring across similar structure instances or by using lexical resources such as WordNet and Levin (1993)'s verb classes. Most of these approaches implement a filtering step, in which inadequate learned structures are discarded on frequency-based grounds.

In this paper, we propose an alternative approach to the problem of determining verb argument structures. We present an unsupervised method for learning the argument structures, modeled over the noisy-channel framework, with the following characteristics:

- It does not assume that the number and order of arguments are known in advance. The argument structures are completely learned from naturally occurring texts.
- The argument structures that we learn are grounded in both lexemes and abstractions (named entities), with the most appropriate abstraction level being automatically determined.
- It ranks competing structures according to their probability.
- It makes use of simple tools, such as part of speech and named entity taggers, that are both widely available and easy to port across languages and domains.

In the rest of the paper, we first describe our statistical model and the algorithms we used to train it (Section 2). We introduce the training data (Section 3) and present a human-based evaluation for a representative sample of verb argument structures that we learn automatically (Section 4). We end with a discussion of the strengths and weaknesses of our model and future work (Section 5).

2 Our Approach

We couch our learning problem in a probabilistic noisy-channel framework. This framework has been widely used in statistical natural language processing[1]. In this

[1] For a more detailed discussion about the noisy-channel model in natural language processing tasks and its characteristics, see Marcu and Popescu (2005).

framework, one concocts a generative story that explains how data of interest comes into existence. For instance, Knight and Marcu (2002)'s generative story shows how short sentences can be mapped into long sentences; Brown et al. (1990, 1993) show how sentences in English are probabilistically mapped into French sentences; Soricut and Brill (2004) show how answers can be mapped into questions. In our model (see Figure 4), the generative story explains how natural language sentences (S) are produced by generating first an abstract argument structure (A) and then mapping this structure into strings. Our generative story goes like this:

1. (a) The head (verb) of the argument structure is first chosen with probability P(v). (b) The number of arguments the verb takes is chosen with probability narg(no_arg | v). (c) Each argument is generated with probability arg(argument | v). Each argument can be either an abstraction/concept (named entity in our case) or a word/lexeme.
2. Once the verb argument structure is generated, a probabilistic parameter phi(N | v) decides the number of extra words/concepts that are going to be eventually produced in the sentence.
3. Each extra word/concept is stochastically generated according to the distribution ew(word).
4. If the generative process produces concepts c (named entities), these are translated into words, with probability t(word | c).

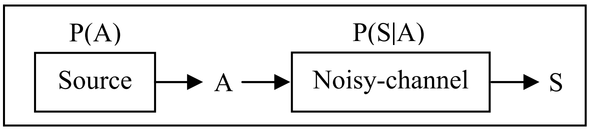

Figure 4. A noisy-channel model for learning verb argument structures

For instance, the sentence *Santa has bought them gifts* can be generated by the following process.

1. (a) The head verb *bought* is first chosen with probability P(*bought*). (b) The verb is associated with 3 arguments with probability narg(3 | *bought*), which are *PERSON1*, *gifts*, and *PERSON2* (c) with probabilities arg(*PERSON1* | *bought*), arg(*gifts* | *bought*) and arg(*PERSON2* | *bought*), respectively. At the end of this sequence, we have available the following verb argument structure: *bought*(*PERSON1*, *gifts*, *PERSON2*).
2. One extra word is added with probability phi(1 | *bought*)
3. which turns out to be the word *has* with probability ew(*has*).
4. The named entities are translated into words: *PERSON1* into *Santa* and *PERSON2* into *them*, with probabilities t(*Santa* | *PERSON1*) and t(*them* | *PERSON2*).

In order to make the training of our model tractable, we make some simplifying assumptions. That is, we assume that the subsequence corresponding to steps 1.a-c happens in one shot: an entire event is generated stochastically with probability event(verb(arg_1,…,arg_n)). Since named entity taggers work at levels of accuracy above 90%, we also assume that it is not necessary to translate concepts into words as part of the generative process – we can pre-tag the sentences used for training with named entity tags and learn argument structures that include such entities directly. From a generative story perspective, this means that we no longer need Step 4 to model the translation of entities into words. Mathematically, these choices simplify our model tremendously. According to the resulting model, the probability of a sentence S is thus given by the following formula:

$$P(S) = \sum_{A} P(S,A) = \sum_{A} P(A) \times P(S|A) = \sum_{A} \text{event}(A) \times \text{phi}(N \mid \text{verb}) \times \prod_{i=1}^{N} \text{ew}(w_i)$$

where A is a possible argument structure, N is the number of extra words/concepts that are generated, and w_i is the i^{th} extra word being generated. In this view, the probability of the sentence P(*Santa/PERSON1 has bought them/PERSON2 gifts*) is event(*bought*(*PERSON1*, *gifts*, *PERSON2*)) × phi(*1* | *bought*) × ew(*has*).

We use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to estimate the parameters of the model (which are uniformly initialized). To restrict the search space and make the training feasible, we assume that a verb can have at most 3 arguments and that arguments can be only open class words (verbs, adjectives, adverbs and nouns – including pronouns). In order to impose these restrictions, we pre-tag the data with a part of speech tagger (Ratnaparki, 1996). Now, we are capable of doing full EM training on our data, as the number of hidden alignments/argument structures that we have to consider for every sentence is reasonable. For example, Figure 5 shows all possible hidden argument structures for the sentence *He has bought them gifts*. The arrows leave from the verb and point to the arguments. The words not pointed to by any arrow are the extra words. For simplicity, the named entity and part of speech tags are not shown.

Figure 5. Possible argument structures in the sentence *He has bought them gifts*

Because we use EM, low probabilities are naturally assigned to uncommon or, hopefully, inadequate argument structures. Therefore, it is not necessary to filter our results in an ad-hoc manner.

3 Data Preparation

From TREC'2002 data collection (Voorhees and Buckland, 2002), we have selected the 1,500 most frequent verbs for training our model. We extracted from TREC'2002 corpus all sentences containing occurrences of these verbs. Since our model is not ready to properly cope with very long sentences (especially those that contain complex verb sentential complements), we filtered out the sentences longer than 10 words. We tagged every sentence using the BBN IdentiFinder named entity tagger (Bikel et al., 1999) and Ratnaparki's (1996) part of speech tagger. On average, we ended up with nearly 1,400 sentences per verb and a total of 14 million words in the collected corpus.

The use of a named entity tagger is not necessary for our model to work; its use, however, enable the model to learn more general argument structures. If entities are not used, the structures we learn are completely lexicalized; if a named entity tagger is used, we expect to learn both lexicalized and generalized verb argument structures. As expected, named entities overcome some of the data sparseness problems and yield argument structures that are more likely than the fully lexicalized ones. It is worth noting that the most appropriate level of abstraction for arguments (lexemes vs. named entities) is learned automatically by the EM algorithm.

Using WordNet concepts for representing the abstraction level is also possible, like many works do. We chose named entities because of the following advantages: the set of entities is more intuitive and small, making the learning process more effective; during tagging, the correct sense of the word is determined.

Other arrangements we did to our data include: all numbers were replaced by the general entity *number*; excepting *it*, *they* and *them*, all personal pronouns were replaced by the entity *person*; *it*, *they* and *them* were considered to be both *person* and the generic entity *thing* (that can be anything but *person*), since they can refer to anything.

Figure 6 shows a sample of our learning data, with entities in bold.

about/IN **money**/NN home/NN water/NN heaters/NNS are/VBP bought/VBN each/DT year/NN

organization/NNP bought/VBD **organization**/NN from/IN **organization**/NN last/JJ year/NN

thing/PRP bought/VBD the/DT outstanding/JJ shares/NNS on/IN **date**/NNP

the/DT cafeteria/NN bought/VBD extra/JJ plates/NNS

Figure 6. Data sample

It is easy to note that some sentences are completely lexicalized, without entities (e.g., the last sentence), while others have several entities. In the first sentence, one can also note an error introduced by BBN IdentiFinder: *8 million* was misclassified as **money**. Such errors should be naturally discarded by EM as valid arguments, since they are not frequent in our corpus.

4 Evaluation and Analysis

To assess the correctness of the verb argument structures we learned automatically, we carried out two experiments with a randomly selected sample of 20 verbs, assuring that it includes low (i.e., rare), medium and high-frequency (i.e., common) verbs in our corpus. The first row in Table 1 shows the selected verbs: "hook", "spin" and "yell" are examples of low-frequency verbs; "raise" and "spend" are examples of medium-frequency verbs; "buy", "die" and "help" are examples of high-frequency verbs.

We compare our results to the results obtained with a baseline algorithm. This algorithm uses a frequency-based method to produce the argument structures: it computes all possible structures that the sentences in our corpus can have, in the same way we show in the example in Figure 5, and ranks the produced argument structures according to their frequencies. Like in our model, the part-of-speech tags and entities are also taken into consideration, i.e., the baseline algorithm is informed about which words can be arguments and is able to learn generalized structures. As will be noted, this baseline algorithm turns out to be very strong.

For the experiments we carried out, for each verb, we kept only the argument structures learned with probabilities above a threshold of 10^{-3} in order to make the evaluation feasible (for some verbs, our model learns hundreds of possible argument structures). Having this, for each verb, we took the same number of structures produced by the baseline algorithm, selecting the most probable ones. This way, we guarantee that the evaluation is fair.

In the first experiment, we wanted to verify how many correct/plausible argument structures were learned by our model in relation to all structures learned. This is a precision measure. We presented the argument structures to three judges (computational linguists) and asked them to independently judge their correctness/plausibility. Each argument structure could be classified as "correct", "wrong" or "can't tell" by each judge: it should be classified as "correct" if the judge could come up with a sentence from the structure; "wrong" in the case it is not possible to come up with a sentence; and "can't tell" when it is not possible to know for sure.

In the second experiment, in order to verify the correspondence of the learned structures to the ones predicted by humans for the verbs, we compared our structures to the ones in PropBank. We computed how many structures in PropBank were learned by our model, observing the number of arguments and their types in each structure and the overall frame. This is, basically, a recall measure. It is important to note that precision was not evaluated in relation to PropBank structures because PropBank is not complete and, as will be discussed here, our model learns argument structures not predicted by this repository.

The same evaluation was carried out for the structures learned by the baseline algorithm.

The 2nd and 3rd columns in Table 1 show the number of sentences used for training our model for each verb and the number of argument structures considered in the experiments. The 4th column in Table 1 shows **P**recision (the average for the three judges) and **R**ecall for each verb. In relation to precision, the annotation agreement between judges was high: the kappa statistic (Carletta, 1996) was 0.69. A kappa figure between 0.6 and 0.8 indicates high agreement. In average, our model achieved 93.7% precision and 73.5% recall, showing good results for low, medium and high-frequency verbs. The 5th column in the table shows the corresponding results for the structures produced by the baseline algorithm: on average, it achieved 81.4% precision and 59.2% recall. The baseline showed to be a strong one, but our model outperformed it. We suspect the good performance of the baseline is explained by the methodology we used to do data collection.

Table 1. Performance of verb argument structure induction algorithm

Verbs	Sentences	Structures	P & R (%) ***Our model***	P & R (%) ***Baseline***
abandon	171	3	100, 50.0	100, 0
aspire	25	3	100, 100	100, 100
avoid	482	6	100, 100	100, 100
buy	2326	44	85.5, 70.0	75.6, 70.0
cause	1301	29	93.0, 100	63.1, 100
collapse	153	4	91.6, 75.0	66.6, 50.0
die	4334	70	85.2, 100	57.5, 100
earn	971	43	88.3, 75.0	76.7, 50.0
expect	2597	64	84.3, 100	70.8, 100
fix	270	18	86.9, 40.0	75.8, 20.0
hate	594	27	91.3, 100	71.5, 100
help	3706	54	89.4, 100	76.4, 100
hook	46	2	100, 33.3	100, 0
issue	955	10	100, 75.0	73.3, 50.0
offer	3071	41	95.8, 20.0	77.9, 20.0
paint	253	5	93.3, 33.3	100, 16.6
raise	1422	63	93.0, 83.3	66.6, 50.0
spend	1560	22	96.9, 100	77.2, 25.0
spin	111	4	100, 66.6	100, 33.3
yell	110	5	100, 50.0	100, 100
Avg.	**1223**	**26**	**93.7, 73.5**	**81.4, 59.2**

We computed the same results for the 10 and 20 most probable structures for each verb in order to verify how the consideration of more low-probability structures interfere in the performance of our model. Table 2 shows the results obtained. As expected, one can note that, as more argument structures we consider, precision decreases and recall increases.

Table 2. Performance of the algorithm for top-10, top-20 and all argument structures

Structures	Our model		Baseline	
	P (%)	R (%)	P (%)	R (%)
Top-10	95.2	63.1	86.6	47.5
Top-20	93.8	65.6	84.8	52.4
All	93.7	73.5	81.4	59.2

We investigated what led to both recall and precision problems. The recall problems, i.e., the inability of the algorithm to induce certain PropBank structures, is explained by the following reasons:

- Some of the argument types found in PropBank structures were not part of our training corpus, and, therefore, they were not learned.
- A few PropBank structures had more than three arguments; our model predicts at most 3 arguments. For instance, PropBank lists the sentence *John killed Mary with a pipe in the conservatory*, for which the words *John*, *Mary*, *pipe* and *conservatory* are arguments, while our model predicts structures with two entities of type *person* as arguments and a third argument being the instrument or the location of the event, but not both together.

There were two main reasons that explained our precision problems; they pertained to improper handling of adverbs and phrasal verbs. Most of times adverbs are adjuncts, instead of arguments, and, therefore, should not be included in argument structures. However, in some cases, the adverbs are too frequent, co-occurring a lot with some verbs, and look essential to the sentence meaning, like in *He asked rhetorically* and *He asked incredulously*. Corroborating this, PropBank includes adverbs in some argument structures. Phrasal verbs are also a nuisance to our model. For instance, from the sentence *He gave up,* the model learns that either *up* is a possible argument for *gave* or that *gave* asks for 1 argument only, ignoring the particle *up* completely.

For exemplifying the learned structures, Figure 7 shows the top 10 structures learned for the verb *buy* with their associated probabilities.

1 buy(organization,organization)	1.20e-01
2 buy(person,number)	8.44e-02
3 buy(person,thing)	7.10e-02
4 buy(organization,thing)	5.63e-02
5 buy(person,organization)	4.28e-02
6 buy(organization,person)	3.51e-02
7 buy(person,house)	1.54e-02
8 buy(person,thing,anyway)	1.54e-02
9 buy(money,money)	1.40e-02
10 buy(organization,organization,date)	8.63e-03

Figure 7. Argument structures for the verb *buy*

It is worth noting the following:

- the 5th and 6th structures are very similar (in the former, a person buys an organization; in the latter, an organization is bought by a person);

- the 7th structure has a lexicalized item (*house*);
- in the 8th structure, there is an error caused by the inclusion of an adverb (*anyway*) in the structure (because it co-occurred enough times with the verb *buy* in the corpus to be learned by the EM algorithm);
- in the 9th structure, there is an error caused by the phrasal verb *buy down* (like in *dollar bought down the yen*) (because the system is not able to identify this as a phrasal verb and ignore it).

For several verbs, our model was able to learn senses and behaviors not listed in PropBank. For instance, for the verb *raise*, our model learned structures for the 'growing' sense of the verb (like in *Peter was raised in a big city*), which is not annotated in PropBank. Our model could also learn many possible behaviors (not listed in PropBank) for the verb *die*, for instance: (a) In *date*, *person* died; (b) *Person* died in *date*; (c) *Person* died in *date* in *location*; (d) *Person* died in *location* in *date*.

5 Conclusion

The experiments reported in this paper make explicit the strengths and weaknesses of our approach. On the positive side, our model is able to yield high accuracy verb argument structures with no annotation effort, using relatively simple language tools. Our model learns both abstract argument structures, which are grounded in named-entity types, and specific structures, which are grounded in the lexicon. Not only does our method find most of the verb argument structures that are already annotated in the PropBank, but it is also able to suggest structures that are not part of this resource.

On the negative side, our model is still not robust enough to properly handle phrasal verbs, adverbs and complex verb sentential complements. Like FrameNet and VerbNet, our model does not explicitly differentiate between obligatory verb arguments from adjuncts. According to our interests, in the way the model works, we also do not distinguish between active and passive constructions in the learned argument structures. However, this is a simple adaptation that could be done by simply distinguishing the sentences types.

A natural extension of this work consists in complementing the argument structures with more information, e.g., thematic roles and syntactic realization of the arguments. However, these extensions require the use of more sophisticated tools (like syntactic and semantic parsers) to identify the arguments roles and syntactic realizations, making the proposed model less language independent.

The usefulness of a repository of verb argument structures is unquestionable. It is easy to imagine how the learned structures can be used in a variety of natural language generation applications (summarization and machine translation, for example) to assess whether the generated outputs are consistent with a set of pre-learned structures. For instance, if such a system generates text that subsumes an inconsistent structure, that text is probably semantically ill-formed.

This paper presented a first investigation on the statistical modeling of argument structures learning in the noisy-channel framework. All the detected limitations and the improvement possibilities to the model should be investigated in future work, as well as the use of the learned argument structures in natural language applications. Other models with different generative stories should also be tested.

Acknowledgments

The authors are grateful to FAPESP, CAPES, CNPq and Fulbright Commission for supporting this work.

References

Baker, C.F.; Fillmore, C.J.; Lowe, J.B. (1998). The Berkeley FrameNet project. In the *Proceedings of COLING/ACL*, pp. 86-90, Montreal.

Bikel, D.M.; Schwartz, R.; Weischedel, R.M. (1999). An Algorithm that Learns What's in a Name. *Machine Learning* (Special Issue on NLP).

Brent, M.R. (1991). Automatic acquisition of subcategorization frames from untagged text. In the *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 209-214, Berkeley, CA.

Briscoe, T. and Carroll, J. (1997). Automatic extraction of subcategorization from corpora. In the *Proceedings of the 5th ANLP Conference*, pp. 356-363, Washington, D.C.

Brown, P.; Cocke, J.; Della Pietra, S.; Della Pietra, V.; Jelinek, F.; Lafferty, J.; Mercer, R.; Roossin, P. (1990). A statistical approach to machine translation. *Computational Linguistics*, Vol. 16, N. 2, pp. 79-85.

Brown, P.; Della Pietra, S.; Della Pietra, V.; Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, N. 2, pp. 263-311.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, Vol. 22, N. 2, pp. 249–254.

Dempster, A.P.; Laird N.M.; Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Ser B, Vol. 39, pp. 1-38.

Framis, F.R. (1994). An experiment on learning appropriate selection restrictions from a parsed corpus. In the *Proceedings of the International Conference on Computational Linguistics*, Kyoto, Japan.

Gildea, D. (2002). Probabilistic Models of Verb-Argument Structure. In the *Proceedings of the 17th International Conference on Computational Linguistics*.

Gomez, F. (2004). Building Verb Predicates: A Computational View. In the *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pp. 359-366, Barcelona, Spain.

Green, R.; Dorr, B.J.; Resnik, P. (2004). Inducing Frame Semantic Verb Classes from WordNet and LDOCE. In the *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pp. 375-382, Barcelona, Spain.

Grishman, R. and Sterling, J. (1992). Acquisition of selectional patterns. In the *Proceedings of the International Conference on Computational Linguistics*, pp. 658-664, Nantes, France.

Grishman, R. and Sterling, J. (1994). Generalizing Automatically Generated Selectional Patterns. In the *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, Japan.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In the *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas.

Kipper, K.; Dang, H.T.; Palmer, M. (2000). Class-based Construction of a Verb Lexicon. In the *Proceedings of AAAI 17th National Conference on Artificial Intelligence*. Austin, Texas.

Knight, K. and Marcu, D. (2002). Summarization beyond sentence extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence*, Vol. 139, N. 1.

Korhonen, A. (2002). Semantically Motivated Subcategorization Acquisition. In the *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon*, pp. 51-58.

Lapata, M. (1999). Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In the *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 394-404.

Levin, B. (1993). *Towards a lexical organization of English verbs*. Chicago University Press, Chicago.

Manning, C. (1993). Automatic acquisition of a large subcategorization dictionary from corpora. In the *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 235-242, Columbus, Ohio.

Marcu, D. and Popescu, A.M. (2005). Towards Developing Probabilistic Generative Models for Reasoning with Natural Language Representations. In the *Proceedings of the 6th International Conference on Computational Linguistics and Text Processing* (Lecture Notes in Computer Science 2406 Springer 2005, ISBN 3-540-24523-5). Mexico City, Mexico.

Marcus, M.; Santorini, B.; Marcinkiewicz, M.A. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, Vol. 19, N. 2, pp. 313-330.

Marcus, M. (1994). The Penn Treebank: A revised corpus design for extracting predicate-argument structure. In the *Proceedings of the ARPA Human Language Technology Workshop*, Princeton, NJ.

McCarthy, D. (2000). Using semantic preferences to identify verbal participation in role switching alternations. In the *Proceedings of the 1st NAACL*, pp. 256-263, Seattle, Washington.

Merlo, P. and Stevenson, S. (2001). Automatic Verb Classification Based on Statistical Distributions of Argument Structure. *Computational Linguistics*, Vol. 27, N. 3.

Ratnaparki, A. (1996). A Maximum Entropy Part-Of-Speech Tagger. In the Proceedings of the Empirical Methods in Natural Language Processing Conference, University of Pennsylvania.

Resnik, P. (1992). Wordnet and distributional analysis: a class-based approach to lexical discovery. In the *Proceedings of AAAI Workshop on Statistical Methods in NLP*.

Rooth, M.; Stefan, R.; Prescher, D.; Carroll, G.; Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In the *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 104-111, College Park, Maryland.

Sarkar, A. and Zeman, D. (2000). Automatic extraction of subcategorization frames for Czech. In the *Proceedings of the 18th International Conference on Computational Linguistics*.

Sarkar, A. and Tripasai, W. (2002). Learning Verb Argument Structures from Minimally Annotated Corpora. In the *Proceedings of the 19th International Conference on Computational Linguistics*.

Soricut, R. and Brill, E. (2004). Automatic Question Answering: Beyond the Factoid. In the *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*.

Voorhees, E.M. and Buckland, L.P. (eds.) (2002). *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference* (TREC 2002). Department of Commerce, National Institute of Standards and Technology.