

# Domain Adaptation of a Dependency Parser with a Class-Class Selectional Preference Model

Raphael Cohen\*

Yoav Goldberg\*\*

Michael Elhadad

Ben Gurion University of the Negev  
Department of Computer Science  
POB 653 Be'er Sheva, 84105, Israel

{cohenrap,yoavg,elhadad}@cs.bgu.ac.il

## Abstract

When porting parsers to a new domain, many of the errors are related to wrong attachment of out-of-vocabulary words. Since there is no available annotated data to learn the attachment preferences of the target domain words, we attack this problem using a model of selectional preferences based on domain-specific word classes. Our method uses Latent Dirichlet Allocations (LDA) to learn a domain-specific Selectional Preference model in the target domain using un-annotated data. The model provides features that model the affinities among pairs of words in the domain. To incorporate these new features in the parsing model, we adopt the co-training approach and retrain the parser with the selectional preferences features. We apply this method for adapting Easy First, a fast non-directional parser trained on WSJ, to the biomedical domain (Genia Treebank). The Selectional Preference features reduce error by 4.5% over the co-training baseline.

## 1 Introduction

Dependency parsing captures a useful representation of syntactic structure for information extraction. For example, the Stanford Dependency representation has been used extensively in domain-specific relation extraction tasks such as BioNLP09 (Kim, Ohta et al. 2009) and BioNLP11 (Pyysalo, Ohta et al. 2011). One obstacle to widespread adoption of such syntactic representations is that parsers are generally trained on a specific domain (typically WSJ news data) and it has often been observed that the accuracy of dependency parsers drops significantly when used in a domain other than the training domain.

\*Supported by the Lynn and William Frankel Center for Computer Sciences, Ben Gurion University

\*\*Current affiliation: Google Inc.

Domain adaptation for dependency parsing has been explored extensively in the CoNLL 2007 Shared Task (Nivre, Hall et al. 2007). The objective in this task is to adapt an existing parser from a source domain in order to achieve high parsing accuracy on a target domain in which no annotated data is available. Common approaches include self-training (McClosky, Charniak et al. 2006), using word distribution features (Koo, Carreras et al. 2008) and co-training (Sagae and Tsujii 2007). Dredze *et al.* (Dredze, Blitzer et al. 2007) explored a variety of methods for domain adaptation, which consistently showed little improvement and concluded that domain adaptation for dependency parsing is indeed a hard task. Typically, parsing accuracy drops from 90+% in-domain to 80-84% in the target domain.

When porting parsers to the target domain, many of the errors are related to wrong attachment of out-of-vocabulary words, *i.e.*, words which were not observed when training on the source domain. Since there is not sufficient annotated data to learn the attachment preferences of the target domain words, we attack this problem using a model of selectional preferences based on domain-specific word classes.

Selectional preferences (SP) describe the relative affinity of arguments and head of a syntactic relation. For example, in the sentence: “*D3 activates receptors in blood cells from patients*”, the preposition “*from*” may be attached to either “*cells*” or “*receptors*”. However, the head word “*cells*” has greater affinity to “*patients*” than the candidate “*receptors*” would have towards “*patients*”. Note that this preference is highly context-specific.

Several methods for learning SP (not in the context of domain adaptation) have been proposed. Commonly, these methods rely on learning semantic classes for arguments and learning the preference of a predicate to a semantic class. These semantic classes may be derived from manual knowledge bases such as WordNet or FrameNet, or semantic classes learned from large corpora. Recently, Ritter *et al.* (2010) and

Séaghdha (2010) both present induction methods of SP of verb-arguments using LDA (Blei, Ng et al. 2003). Hartung and Frank (2011) extended the LDA-based approach to learning preference for adjective-noun phrases.

In this work, we tackle the task of domain adaptation by developing a domain-specific SP model. Our initial observation is that parsers fail on the target domain when trying to attach domain-specific words not seen during training. We observe as many as 15% of the words are unknown when applying a WSJ-trained parser on Genia and PennBioIE data, compared to only 2.5% in-domain. Parsers trained on the source domain cannot learn attachment preferences for such words. Our motivation is, therefore, to attempt to learn attachment preferences for domain specific words using un-annotated data. Specifically, we focus on acquiring a domain-specific SP model.

Our approach consists of using the low-accuracy source-domain parser on large quantities of in-domain sentences. We extract from the resulting parse trees a collection of syntactically related pairs of words. We then train an LDA model over these pairs of words and derive a domain-specific model of lexical affinities between pairs of words. We finally re-train a parser model to exploit this domain-specific data. To this end, we use the approach of co-training, which consists of identifying reliable parse trees in the target domain in an unsupervised manner using an ensemble of two distinct parsers, and extending the annotated training set with these reliable parse trees. Co-training alone significantly reduces the proportion of unknown words in the re-trained parser – in the extended co-training dataset, we observe that the unknown words rate drops from 15% to 4.5%. Data sparseness, however, remains an issue: 1/3 of the domain-specific words added to the model by co-training appear only once in the extended training set, and we observe that many of the attachment errors are concentrated in a few syntactic configurations (*e.g.*, head(V or N)-prepobj, N-N or head(N)-Adj). We extend co-training by introducing our SP model, which is class-based and specific to these difficult syntactic configurations.

Our method reduces error in the Genia Treebank (Tateisi, Yakushiji et al. 2005) by 3.5% over co-training. Introducing additional distributional lexical features (Brown clusters learned in-domain), further reduces error to a total 4.5% reduction. Overall, our parser achieves an accuracy of 83.6% UAS on the Genia domain without annotated data in this domain.

## 2 Our Approach

To understand the difficulty of domain adaptation, we applied our parser trained on the WSJ news domain to the Genia and measured observed errors. Most of the errors were found in a small set of syntactic configurations: verb-prep-noun, noun-adjective, noun-noun (together these relations make up 32 % of the errors).

For example: in “*nuclear factor-kappa-B DNA-binding activity*” the parser chooses “*factor-kappa-B*” as the head of “*nuclear*” instead of “*activity*”. We observe that these errors involve domain-specific vocabulary, and are difficult to disambiguate for non-expert humans as well. Accordingly, we try to acquire a domain-specific model of word-pairs affinities. Our parsing model (EasyFirst) allows us to use such bi-lexical features in an efficient manner. Because of data sparseness, however, we aim to acquire class-based features, and decide to model these lexical preferences using the LDA approach.

Our method proceeds in two stages:

1. Learn selectional preferences from an automatically parsed corpus using LDA on selected syntactic configurations
2. Integrate the preferences into the parsing model as new features using co-training.

### 2.1 Learning Selectional Preferences

Following (Ritter, Mausam et al. 2010) and (Séaghdha 2010), we model lexical affinity between words in specific syntactic configurations using LDA. Traditionally, LDA learns a set of “topics” from observed documents, based on observed word co-occurrences. In our case, we form artificial documents, which we call syntactic contexts, by collecting head-daughter pairs from parse trees. A syntactic context is constructed for each head word, which contains the related words to which it was found attached.

In the collection process, we identify two syntactic configurations that yield high error rates: *head-prep-noun* and *noun-adj*. We collect two types of syntactic contexts: the preposition contexts contain the set of nouns related to the head through any preposition and the adjective contexts contain the set of adjectives directly related to the head noun. We then learn an LDA model on each of these contexts collections. We use Mallet (McCallum 2002) to learn topic models with hyper-parameter optimization (Wallach, Mimno et al. 2009). The optimal number of topics is selected empirically based on model fit to held-out data.

Source	Relation Type	Semantic Class	Arguments	Predicates
BLLIP	Arg → Prep → Predicate	Show Business	actors clips soundtrack genre taping characters roles immortalized starred costumes premise screening featured performances poster trumpeted star retrospective clip script	film show movie films movies shows television series stage theater program production version music hollywood broadway
BLLIP	Arg → Prep → Predicate	Sports	quarterbacks starters pitcher pitchers quarterback coaching receiver linebackers cornerback outfielder baseman fullback	team game league teams games time field players years baseball year rules nfl seasons level player leagues nba club history school state
BLLIP	Arg → Prep → Predicate	Work Position	jockeying groom groomed relegate relieved unwinding jockeyed selected selecting appointing disqualify named	job post position draft positions candidate team one jobs which role posts successor
Genia	Arg → Prep → Predicate	Cell-cycle process	stages stage process steps committed block regulator acquire switch points needed directs determinant il-21 proceeds arrest regulators relate d3	differentiation development activation maturation cycle hematopoiesis infection commitment lymphopoiesis stage lineage selection erythropoiesis cascade
Genia	Arg → Prep → Predicate	Cells and growing conditions	supernatants co-culture co-cultured replication medium surface chemotaxis supernatant beta migration cocultured cultures hyporesponsiveness	cell monocyte lymphocyte pbmc macrophage line blood neutrophil cd dc leukocyte t eosinophil fibroblast platelet keratinocyte
Genia	Adjective → Noun	Protein activity and regulation	factor-induced tnfalpa-induced agonist- induced thrombin-induced il-2-induced factor- alpha-induced il-1beta-induced cd40-induced rankl-induced augmented il-4-induced	expression activation production phosphorylation response proliferation activity binding secretion apoptosis differentiation translocation release signaling adhesion synthesis generation

Table 1 High affinity classes in the Class-Class Selectional Preferences model extracted with LDA. Classes 1-5 are from preposition head/object pairs (e.g. “groomed for position” fits the third topic) and class 6 are adjective modifier pairs. Classes 1-3 are from Bllip (un-annotated WSJ corpus) (Charniak, Blaheta et al. 2000) while classes 4-6 are from a corpus composed of Medline abstracts from the Genia (see section 5.1). Class 4 contains arguments and predicates concerning cell-cycle process. In class 5 arguments are cell growing conditions and predicates are types of cells.

The resulting topics represent latent semantic classes of the daughter words. We define a measure of shared affinity between a head word  $h$  and a candidate daughter word  $d$  (in a given configuration)  $s$ :  $Affinity(h, d) = \sum_{c \in \mathcal{T}_{topics}} P(c|h) * P(c|d)$  where  $P(c|h)$  is the predicted probability of topic  $c$  given the syntactic context associated to head word  $h$ . That is, when we apply the LDA model on the syntactic context of  $h$ , we assign topics to each of the associated daughter words and count their proportion. Note that this affinity measure may predict a non-zero affinity to a pair  $(h, d)$  even though this word pair has never been observed. The result is a class-class SP model with reduced dimensionality compared to word-word models based for example on PMI. Table 1 lists examples of learned topics. Note that these topics are high-quality semantic clusters that reflect domain semantics, with marked differences between the news and bio-medical domains.

## 2.2 Co-training to exploit domain features

At this stage, we have acquired a domain-specific model of word affinity that exploits semantic classes and depends on specific syntactic configurations (*head-prep-obj* and *noun-adj*). We now attempt to exploit this model to adapt our source parser to the target domain. To this end, we want to re-train the parser using new features based on the SP model in addition to the original features. We use the framework of co-training to achieve this goal (Sagae

and Tsujii 2007): we use two different parsers: Easy-First (Goldberg and Elhadad 2010) and MALT (Nivre, Hall et al. 2006) trained on the same WSJ source domain. We apply these two parsers on a large set of target-domain sentences. We select those sentences where the 2 parsers agree (produce identical trees) and add them to the original source-domain training set. We thus obtain an extended training set with many in-domain samples. We can now re-train the parser using the new SP features.

## 2.3 SP as features for the Easy First parser

We use the deterministic non-directional Easy-First parser for re-training. This parser incrementally adds edges between words starting with the easier decisions before continuing to difficult ones. Simple structures are first created and their information is available when deciding how to connect complex ones. Easy-First operates in  $O(n \log n)$  time compared to  $O(n^3)$  of graph-based parsers such as MST (McDonald, Pereira et al. 2005).

As a baseline we use the features provided in the Easy-First distribution. We extend these features with pair-wise affinity measures based on our SP model. The affinity measure ranges from 0 to 1. We bin this measure into (*low*, *medium*, *high*, *very-high*) binary features. When attaching a preposition to its parent, we add one more feature: the affinity of the head candidate with the preposition’s daughter (the *pobj*). In addition to these pair-wise features, we also

introduce features that correspond to the latent topic class of the words according to each of the 2 acquired LDA models (this introduces one binary feature for each topic). These latent semantic class features are similar in nature to distributional lexical features as used in (Koo, Carreras et al. 2008).

The EasyFirst parser combines partial trees bottom-up. When deciding whether to attach the partial tree "from patients" to either "cells" or "receptors", we compute the affinities of "cells/patients" and "receptors/patients". Our model produces features indicating *medium* affinity for "receptors from patients" and a *high* affinity for *cells from patients*".

### 3 Experiments and Evaluation

#### 3.1 Genia Treebank

The Genia Treebank (Tateisi, Yakushiji et al. 2005) contains 18K sentences from the biomedical domain, transformed into dependency trees<sup>1</sup> using (De Marneffe, MacCartney et al. 2006)<sup>2</sup>. The corpus contains 2.3K sentences longer than 40 tokens that were excluded from the evaluation. The treebank was divided into test and development sets of equal size.

We created an un-annotated corpus of 200K sentences by querying Medline with the same query terms used to create Genia. We used the Genia POS Tagger on this dataset (Tsuruoka, Tateishi et al. 2005). The corpus was parsed with Easy-First and MALT (arc-eager, polynomial) to create co-training data, yielding 21K sentences with 100% agreement.

The parsed corpus of 200K sentences was used to produce selectional preference models for *adjective-nouns*, with 200 topics, and for *head-prep-object* with 300 topics. We used word lemmas for each pair when preparing syntactic contexts for LDA training (see Table 2).

Relation	# Pairs	# Daughter	# Heads
preposition	360,041	1,727	2,391
adjective	384,347	1,570	2,003

Table 2. Statistics for the training data of the SP model.

#### 3.2 Coverage

Many of the features learned in training a parser are lexicalized; this is an important factor in the drop in accuracy when parsing in a new domain.

To understand the nature of the contribution of the features learned by our SP model, we calculated the coverage of the features acquired in two unsupervised methods: Brown clustering and our SP classes. We

<sup>1</sup> We use the PTB version of Genia created by Illes Solt.

<sup>2</sup> We convert using the Stanford Parser bundle.

count the number of tokens in the Treebank which gain a feature at training time (we ignore punctuation, coordination and preposition tokens). Our SP model covers 53% of the tokens in the test set. Brown clusters calculated with the implementation of Liang (2005) achieve coverage of 73%. Brown clusters features are also class-based distributional features based on n-gram language models, but do not take into account syntactic configurations.

#### 3.3 Adaptation Evaluation

We use a number of baselines for the adaptation task. Three parsers were evaluated on the target domain: Easy-First, MST second order and MALT arc-eager with a polynomial kernel. We report UAS scores of trees of length < 40 without punctuation.

The first baseline setting for each parser is the model trained on WSJ sections 2-21. The second baseline we report is co-training using WSJ 2-21 combined with the 21K full agreement parse trees extracted from Medline, but without new features.

Parser	Training Data	Features	UAS (Exact Match)	
MST	WSJ 2-21		79.6 (10)	
MALT	WSJ 2-21		81.1 (16.6)	
Easy-First	WSJ 2-21		80.5 (12.3)	
MST	Co-Training		81.3 (14.1)	
MALT	Co-Training		82.1 (16.5)	
Easy-First	Co-Training		82.8 (16.2)	
Easy-First	Co-Training	+Brown Clusters	83.1 (17)	+0.3
Easy-First	Co-Training	+SP-Lexicalized	83.0 (16.9)	+0.2
Easy-First	Co-Training	+SP-Lexicalized +SP-Classes	83.4 (16.6)	+0.6
Easy-First	Co-Training	+SP-Lexicalized +SP-Classes +Brown Clusters	<b>83.6 (17.2)</b>	+0.8
Easy-First	GeniaTB Dev		89.8 (28.6)	

Table 3. Accuracy for different parser settings on Genia test set. The best performing adapted model trains with co-training data and combines SP and Brown clusters as features.

In Table 3, we see that the combined SP-Features improved the co-training baseline by 0.6%, a significant error reduction of 3.5% (p-value < 0.01).

We list improvement when introducing only pair-wise SP features, and when adding SP-based semantic classes. The effect is also additive with the Brown clusters features, producing an improvement of 0.8% when combined (error reduction of 4.5%).

To evaluate the model adapted for Genia on the general biomedical domain, we used the PennBioIE Treebank. This dataset contains 6K sentences from different biomedical domains. We compared 3 models (see Table 4):

1. Easy-First, MALT and MST trained on WSJ.
2. Easy-First with co-training on Genia.

### 3. Easy-First with co-training on Genia with Selectional Preference features.

Domain adaptation to Genia carried over to the closely related PennBioIE dataset, demonstrating the generalization capability of the method.

Parser	Training Data	Features	UAS	
MALT	WSJ 2-21		78.8	
MST	WSJ 2-21		81.4	
Easy-First	WSJ 2-21		79.8	
Easy-First	Co-Training		81.9	
Easy-First	Co-Training	+SP-Lexicalized	<b>82.2</b>	+0.3
		+SP-Classes		
		+Brown Clusters		

Table 4. Accuracy of parsers on PennBioIE Treebank.

### 3.4 Error Analysis

We compare the parser using the SP pair-wise features for preposition attachment to the co-trained baseline on Genia. The overall accuracy of the parser is improved by 0.2%. However, the two models agree only on 90% of the edges, indicating the new SP features play a very active role when parsing.

For “*E3330 inhibited this induced promoter activity in a dose-dependent manner*”, the co-trained parser chose “*activity*” as the head of “*in*” instead of “*inhibited*”. The affinity feature in our model for (“*inhibited*”, “*manner*”) shows affinity of *high* (40-60%) compared to *low* (5-20%) for the wrong pair (“*activity*”, “*manner*”). The same change occurs for “*LysoPC attenuates activation during inflammation and athero-sclerosis*”, where the improved model prefers the pair (“*attenuates*”, “*inflammation*”) to the pair (“*activation*”, “*inflammation*”) which was chosen by the co-trained model.

The modest overall improvement is due to errors introduced by the new model. In “*Tissue obtained from ectopic pregnancies may identify the mechanism of trophoblast invasion in ectopic pregnancies*”, the correct governor of “*in*” is “*invasion*”. However, the SP model ranks the affinity of (“*invasion*”, “*pregnancies*”) lower than that of (“*mechanism*”, “*pregnancies*”).

Most of the improvement of the full SP model (+0.6%) comes from an improvement in the *N-N* relation from 83% to 84.9% (11% error reduction), this improvement is due to semantic classes features learned on the relations of noun-adjective and head-prep-pobj.

### 3.5 Effect on NER

Since most of the improvement comes from the *N-N* relation, we expect improvement for downstream applications such as Named Entity Recognition, a basic task frequently used in the biomedical domain.

We use the portion of the Genia Treebank covered by the Genia NER corpus (Kim, Ohta et al. 2004). We expect the inner tokens of a named entity to be connected by relation of *N-N* or *N-Adj*. We evaluate the accuracy of these two relations for NE tokens. The Easy-First with co-training baseline produces accuracy of 82.9% on this specific set of relations, improved by the SP model to 84.4%, a reduction in error of 8.7%.

## 4 Related Work

### 4.1 Learning of Selectional Preference

Preference of predicate-argument pairs has been studied in depth with a number of approaches. Resnik (1993) suggested a class-based model for preference of predicates combining WordNet classes with mutual information techniques for associating an argument with a predicate class from WordNet.

Another approach models words in a corpus as context vectors (Erk and Pado 2008; Turney and Pantel 2010) for discovering predicate or argument classes using large corpora or the Web.

Recently, semantic classes were successfully induced using LDA topic modeling. These methods have shown success in modeling verb argument relationship to a single predicate (Ritter, Mausam et al. 2010) or a predicate pair (Séaghdha 2010), as well as for adjective-noun preference (Hartung and Frank 2011).

### 4.2 Learning SP for improving dependency parsing

The argument-predicate choice learned in SP is directly related to the decision of creating an edge between them in a parse tree. Van Noord (2007) modeled verb-noun preferences using pointwise mutual information (PMI) using an automatically parsed corpus in Dutch. Association scores of pairs were added as features improving the accuracy significantly from 87.4% to 87.9%.

Nakov and Hearst (Nakov and Hearst 2005) focused on resolving PP attachments and coordination. They used co-occurrence counts from web queries in order to estimate selectional restrictions.

Zhou et al. (2011) used N-gram counts from Google search and Google VI to deduce word-word attachment preferences. They used these counts in a pair-wise mutual information (PMI) scheme as features for improving parsing in the News domain (WSJ) and adaptation for biomedical domain. Their evaluation showed improvement of 1% on WSJ

section 23 over the vanilla MST parser and a significant increase in the domain adaptation problem.

### 4.3 Domain adaptation of dependency parsing

Domain adaptation for dependency parsing has been studied mostly in regard to the CoNLL 2007 shared task (Nivre, Hall et al. 2007). Both of the leading methods included learning from a parser ensemble. Attardi *et al.*'s (2007) used a weak parser in order to identify common parsing errors and overcome those in the training of a stronger parser. Sagae and Tsujii (2007) used two different parsers to parse un-annotated in-domain data and used the trees where the two parsers agreed to augment the training corpus. Dredze *et al.* (2007) approached the "closed" problem, *i.e.*, without using additional un-annotated data. They used the PennBioIE Treebank and applied a number of adaptation techniques: (1) features concerning NPs such as chunking information and frequency; (2) word distribution features; (3) features encoding information from diverse parsers; (4) target focused learning – giving greater weight in training to sentences which are more likely in a target domain language model. These methods have not improved accuracy over the baseline of the MST parser (McDonald, Pereira et al. 2005) trained on WSJ.

## 5 Conclusion

Learning class-class selectional preferences from a large in-domain corpus assists dependency parsing significantly. We have suggested a method for learning selectional preference classes for a specific domain using an existing parser and a standard implementation of LDA topic modeling. The SP model can be used for estimating the affinity between a pair of tokens or simply as a feature of semantic class association. This approach is faster when querying the model for the affinity of a pair of words than a PMI model suggested by Zhou *et al.* (2011). While covering fewer tokens in the target test set than Brown clusters, the method achieved a higher improvement of parsing performance. Furthermore, some of the improvement was additive and reduced UAS error by 4.5% compared to a strong co-training baseline.

## 6 References

Attardi, G., F. Dell'Orletta, et al. (2007). Multilingual dependency parsing and domain adaptation using DeSR. ACL.

Blei, D. M., A. Y. Ng, et al. (2003). "Latent dirichlet allocation." JMLR 3: 993-1022.

Charniak, E., D. Blaheta, et al. (2000). "Blip 1987-89 wsj corpus release 1." LDC.

De Marneffe, M. C., B. MacCartney, et al. (2006). Generating typed dependency parses from phrase structure parses. LREC.

Dredze, M., J. Blitzer, et al. (2007). Frustratingly hard domain adaptation for dependency parsing. CoNLL 2007.

Erk, K. and S. Pado (2008). A structured vector space model for word meaning in context. EMNLP 2008: 897-906.

Goldberg, Y. and M. Elhadad (2010). An efficient algorithm for easy-first non-directional dependency parsing. NAACL 2010: 742-750.

Hartung, M. and A. Frank (2011). Exploring Supervised LDA Models for Assigning Attributes to Adjective-Noun Phrases. ACL.

Kim, J.-D., T. Ohta, et al. (2009). Overview of BioNLP'09 shared task on event extraction. Current Trends in Biomedical NLP, ACL: 1-9.

Kim, J.-D., T. Ohta, et al. (2004). Introduction to the bio-entity recognition task at JNLPBA. Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications. Geneva, Switzerland, Association for Computational Linguistics: 70-75.

Koo, T., X. Carreras, et al. (2008). Simple semi-supervised dependency parsing. ACL 2008: 595-603.

Liang, P. (2005). Semi-supervised learning for natural language, Massachusetts Institute of Technology.

McCallum, A. K. (2002). "Mallet: A machine learning for language toolkit."

McClosky, D., E. Charniak, et al. (2006). Effective self-training for parsing, ACL.

McDonald, R., F. Pereira, et al. (2005). Non-projective dependency parsing using spanning tree algorithms. EMNLP: 523-530.

Nakov, P. and M. Hearst (2005). Using the web as an implicit training set: application to structural ambiguity resolution. EMNLP, Association for Computational Linguistics: 835-842.

Nivre, J., J. Hall, et al. (2007). The CoNLL 2007 Shared Task on Dependency Parsing, CoNLL 2007. s. **915-932**.

Nivre, J., J. Hall, et al. (2006). Maltparser: A data-driven parser-generator for dependency parsing.

Noord, G. v. (2007). Using self-trained bilinear preferences to improve disambiguation accuracy. 10th International Conference on Parsing Technologies, ACL: 1-10.

Pyysalo, S., T. Ohta, et al. (2011). "Overview of the Entity Relations (REL) supporting task of BioNLP 2011." ACL HLT 2011 1(480): 83.

Ritter, A., Mausam, et al. (2010). A latent dirichlet allocation method for selectional preferences. ACL 2010: 424-434.

Sagae, K. and J.-i. Tsujii (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. EMNLP-CoNLL 2007: 1044-1050.

Séaghdha, D. (2010). Latent variable models of selectional preference. ACL 2010: 435-444.

Tateisi, Y., A. Yakushiji, et al. (2005). Syntax Annotation for the GENIA corpus. ACL.

Tsuruoka, Y., Y. Tateishi, et al. (2005). "Developing a robust part-of-speech tagger for biomedical text." AII: 382-392.

Turney, P. D. and P. Pantel (2010). "From frequency to meaning: Vector space models of semantics." JAIR 37(1): 141-188.

Wallach, H., D. Mimno, et al. (2009). "Rethinking LDA: Why priors matter." NIPS 22: 1973-1981.

Zhou, G., J. Zhao, et al. (2011). Exploiting web-derived selectional preference to improve statistical dependency parsing. ACL.