Subdomain Sensitive Statistical Parsing using Raw Corpora

Barbara Plank*, Khalil Sima'an†

* Alfa Informatica, Faculty of Arts
University of Groningen, The Netherlands
b.plank@rug.nl

† Language and Computation, Faculty of Science
University of Amsterdam, The Netherlands
simaan@science.uva.nl

Abstract

Modern statistical parsers are trained on large annotated corpora (treebanks). These treebanks usually consist of sentences addressing different subdomains (e.g. sports, politics, music), which implies that the statistics gathered by current statistical parsers are mixtures of subdomains of language use. In this paper we present a method that exploits raw subdomain corpora gathered from the web to introduce subdomain sensitivity into a given parser. We employ statistical techniques for creating an ensemble of domain sensitive parsers, and explore methods for amalgamating their predictions. Our experiments show that introducing domain sensitivity by exploiting raw corpora can improve over a tough, state-of-the-art baseline.

1. Motivation

Current state-of-the-art statistical parsers are trained on large syntactically annotated corpora (treebanks) and their parameters are estimated to reflect properties of the training data. Usually, a treebank/corpus consists of language use concerning a range of topics. For example, as observed by (Kneser and Peters, 1997), subdomains like ”politics, stock market, financial news etc. can be found“ in the Wall Street Journal (WSJ) Penn Treebank (PT) (Marcus et al., 1993). Hence, for a statistical parser trained on such a treebank the statistics gathered are averages over different subdomains. By definition, averages smooth-out the statistical differences between the individual subdomains and weaken the biases in the model.

The present paper describes a method for incorporating subdomain sensitivity into an existing state-of-the-art parser (Collins, 1997; Bikel, 2002) in order to improve its predictions. The main idea is to create and combine an ensemble of subdomain sensitive parsers (each for a different subdomain) without the need for further manual annotation of subdomain data. We exploit unannotated, subdomain specific corpora gathered from the web, for re-weighting the original treebank trees to reflect subdomain statistics, and employ that for training individual subdomain sensitive parsers. In what follows we describe first our re-weighting function, followed by methods for combining the subdomain sensitive parsers. Finally, we exhibit the encouraging results of experiments against the baseline state-of-the-art parser.

2. Subdomain Sensitive Parsing

To exploit domain-information and create an ensemble of subdomain sensitive parsers, one approach, used in (Bod, 1999), is to partition the given training treebank into disjoint subtreebanks, each addressing one subdomain. We think that it is often not that straightforward (sometimes even impossible) to partition the training data along clear-cut subdomain borderlines. Furthermore, this approach leads to sparse data problems.

In this paper we suggest a more subtle method than partitioning the treebank. For every subdomain, we create a “subdomain specific version” of the original treebank as follows. For every tree in the original treebank we give it a new count/weight that expresses the likelihood it will appear in the given subdomain. This method can be seen as sampling tree-types according to the specific subdomain distribution. We obtain subdomain specific parsers by training a statistical parser on each of the resulting subdomain specific versions separately. In the sequel, we present the method for tree-weighting followed by methods for parser combination.

2.1. Tree Weighting using LMs

Our Tree Weighting approach exploits *Statistical Language Models* induced from raw subdomain-dependent corpora. In more detail, given a set of subdomains and their corresponding raw corpora, we define a weighting function over the set of training instances in the original treebank TB:

1. For each subdomain-specific corpus, induce a Language Model (LM) θ from the raw subdomain corpus.
2. For every tree π_i in the treebank TB, define its *count* under LM θ to be equal to the average per-word count of its yield $y_{[\pi_i]}$ under LM θ:

   $$f_\theta(\pi_i) = f_\theta(y_{[\pi_i]}) = -\log P_\theta(y_{[\pi_i]})/n$$

   where n is the length of yield $y_{[\pi_i]}$ and P_θ is the probability under LM θ. Thus, the count is given by the length-normalized probability of the yield.
3. Let f_θ^{max} be the maximum count of a tree in TB according to θ. The weight w_i assigned to π_i is defined as:

   $$w_i = \text{round}\left\{ \left(\frac{f_\theta^{max}}{f_\theta(\pi_i)} \right)^a \right\} \qquad (1)$$

   where $a \geq 1$ is a scaling constant. In the default setting $a = 1$.

The Tree Weighting function thus defined for each subdomain effectively results in a subdomain-weighted treebank TB_θ. We employ each of these treebanks TB_θ to retrain the original parser, thereby obtaining a subdomain sensitive parser. The relevant details regarding the subdomain corpora, the treebank and parser used in this study are given in the experimental section of this paper. In the next section, we propose techniques on how to combine the domain-dependent parsers.

2.2. Parser Combination Techniques

Given an ensemble of parsers, the question which naturally follows and is addressed in this section is to devise methods for combining them in order to get a single, final result. There are various ways to combine parsers, including methods for combining the outputs of classifiers by voting schemes or even parse-reranking schemes based on log-linear models (Charniak and Johnson, 2005). For our preliminary exploration in parser combination, we choose here two simple methods:

- parser pre-selection (choose a parser for a given input sentence) and
- post-selection (choose any of the output trees of the parsers given the input).

Probabilistically speaking, the two methods can be combined in a Bayesian decision rule which combines $P(dom|s)$, i.e. the (pre-selection) probability of the parser dom given input sentence s, with $P(t|dom, s)$, the (post-selection) probability of the output tree t given the parser dom and input sentence s:

$$\arg\max_t \sum_{dom} P(dom|s)P(t|dom, s)$$

However, it is hard to obtain good estimates of $P(dom|s)$. Furthermore, the different subdomains we work with cannot comply with the theoretical requirement of being disjoint, thereby conflicting with the Bayesian formula.

In this initial work we concentrate on exploring approaches to pre- and post-selection separately using statistical measures. Combining the two in the Bayesian framework is a much harder task and is left for future research.

2.3. Pre-selection: Divergence Model (DVM)

In parser pre-selection, the goal is to select one of the subdomain parsers to parse an input sentence. The sentence probability obtained from a subdomain language model might not be sensitive enough to discriminate between the different subdomains. A suitable measure must weight words that are more typical for a certain subdomain higher than words shared across subdomains. This is the idea behind our proposed *Divergence Model* (DVM).

Given k raw subdomain corpora, and k unsmoothed unigram Language Models $\theta_1, ..., \theta_k$ induced from the corpora, the *divergence* of some subdomain $i \in [1, .., k]$ on word type w from the $k-1$ other subdomains ($j \in [1, .., k] : j \neq i$) is defined as:

$$divergence_i(w) = 1 + \frac{\sum_{j\in[1,..,k]:j\neq i} |\log \frac{p_{\theta_i}(w)}{p_{\theta_j}(w)}|}{(k-1)} \quad (2)$$

where $|X|$ denotes the *absolute value* of X. The border conditions for this function are given by

- if $p_{\theta_i}(w) = 0$ then $divergence_i(w) = 1$, and
- if $p_{\theta_j}(w) = 0$, then redefine $p_{\theta_j}(w) = const$ where $const$ is a number between 0 and the smallest unigram Language Model probability. In our case, we set $const = 10^{-15}$.

We define the divergence of a sentence $w_1^n = w_1, \ldots, w_n$ to be equal to the average per word divergence:

$$s\_divergence_{i\in[1,..,k]}(w_1^n) = \frac{\sum_{x=1}^{n} divergence_i(w_x)}{n} \quad (3)$$

This divergence score is used to select among the available k subdomain parsers.

2.4. Post-selection: Node Weighting including DVM

In parser post-selection, the input sentence is given to all subdomain parsers and the output parse is selected from the set of output parse trees. The technique we introduce combines a constituent-based score through linear interpolation with the sentence-level score as given by the DVM. We will refer to this method as *Node Weighting including the Divergence Model* (NW-DVM).

Given k candidate parse trees π_i, for $1 \leq i \leq k$, output by k subdomain parsers. Let $\sigma(c, \pi_i)$ be a boolean function, which is equal to 1 if tree π_i contains constituent c, and 0 otherwise. The score for a constituent $c \in \pi_i$ is defined as its average occurrence in all k output trees:

$$score(c) = \left[\sum_{i=1}^{k} \frac{\sigma(c, \pi_i)}{k}\right] \quad (4)$$

Given the function $score(c)$, defined in the range [0,...,1], and the divergence score of a sentence as defined in equation 3, a score for a tree π_i is calculated as:

$$score(\pi_i) = (1-\lambda)\left[\sum_{c\in\pi_i} \frac{score(c)}{|\pi_i|}\right] + \lambda * s\_divergence_i(w_1^n) \quad (5)$$

where $|\pi_i|$ denotes the number of constituents in π_i, and $\lambda \in [0, 1]$ is an interpolation parameter set experimentally to balance the contribution of the DVM and the constituent-weighting subterms.

3. Experiments

All experiments were performed using Bikel's parser (Bikel, 2002), an emulation of Collins' Head-lexicalized Probabilistic Context Free Grammar (PCFG) model (Collins, 1997; Collins, 2003). We use the Penn Treebank (PT) Wall Street Journal (WSJ) (Marcus et al., 1993) version 2, with the now 'standard division' (Collins, 1997; Collins, 2003) into training (sections 02-21), test (section 23) and development/dev (section 00) sets.

As concepts constituting possible subdomains within the PT WSJ we assume: FINANCIAL, POLITICS and SPORTS. For the POLITICS subdomain we use the English part of the *Europarl Parallel Corpus* (Koehn, 2005). For the

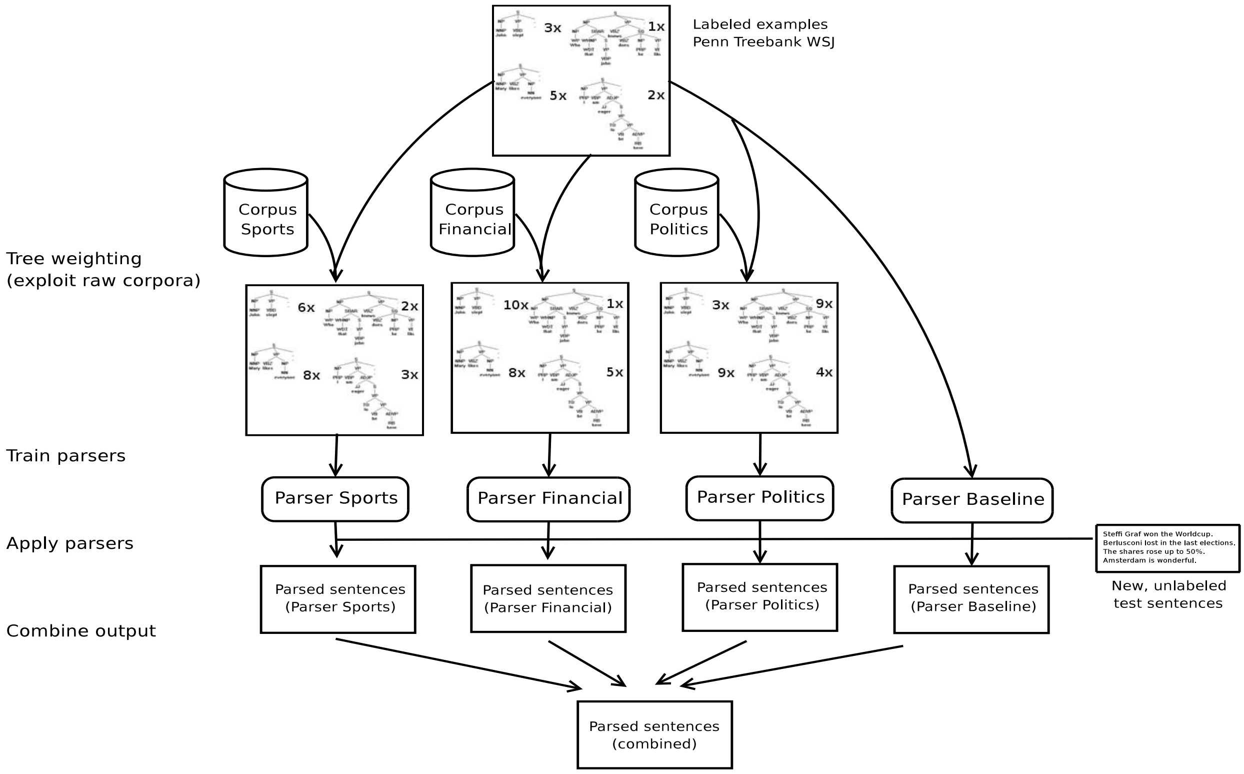
Labeled examples
Penn Treebank WSJ
3x
1x
5x
2x
Corpus Sports
Corpus Financial
Corpus Politics
Tree weighting
(exploit raw corpora)
6x
2x
8x
3x
10x
1x
8x
5x
3x
9x
9x
4x
Train parsers
Parser Sports
Parser Financial
Parser Politics
Parser Baseline
Apply parsers
Steffi Graf won the Worldcup.
Berlusconi lost in the last elections.
The shares rose up to 50%.
Amsterdam is wonderful.
New, unlabeled test sentences
Parsed sentences (Parser Sports)
Parsed sentences (Parser Financial)
Parsed sentences (Parser Politics)
Parsed sentences (Parser Baseline)
Combine output
Parsed sentences (combined)

Figure 1: Summary of the Experimental Design for Sub-domain Aware Parsing

FINANCIAL and SPORTS subdomains, to the best of our knowledge there were no ready-to-use corpora available. Hence, we used Wikipedia (Wikimedia Foundation Inc., 2007) to create domain-specific corpora ourselves. We used Wikipedia's category system to extract relevant articles from the English Wikipedia's dump file, cleaned the articles from Wiki-syntax and segmented them into a one sentence-per-line corpus. The size of the resulting raw domain-specific corpora ranges from 6 to 11 million tokens. For each of the possible subdomains Statistical Language Models (LMs) were estimated and smoothed (using Chen and Goodman's modified Kneser-Ney smoothing) by using the SRI Language Modeling Toolkit[1] (SRILM). Then, we applied our Tree Weighting function (section 2.1.) to create the subdomain specific training data[2] for the subdomain-sensitive parsers. The size of the resulting treebanks is between 127k and 160k training instances.

Figure 1 summarizes our experimental setting. The subdomain-weighted versions of the training treebank TB_θ are created to train the respective parsing model. Our ensemble of parsers consists of a total of four parsers: three domain-dependent parsers, and the baseline parser. The domain-dependent parsers represent the FINANCIAL, POLITICS and SPORTS domains, respectively. The baseline parser is trained on the original treebank, the usual Penn Treebank WSJ sections 02 to 21, and is included in the ensemble to represent the "general" domain REST.

To evaluate parsing performance we use EVALB and the standard PARSEVAL evaluation metrics. Results of parsing sentences of length up to 40 words are reported for both the development (section 00) and test set (section 23).

Before assessing the ability of the parser combination techniques to select better parses, we first gauge in how far the Tree Weighting method (section 2.1.) results in a set of subdomains parsers with complementary capabilities.

3.1. Effectiveness of Subdomain Tree Weighting

To gauge whether subdomain Tree Weighting is giving more varied and useful subdomain specific parsers, we combine the output of the parsers using an optimal decision procedure, i.e. an oracle. Given i candidate parse trees, the oracle is a decision procedure that selects the best tree by measuring the accuracy in F-score[3] against the gold standard tree.

Parser / Combination Technique	LR	LP	F-score
	Section 00 (devset)		
Baseline	89.44	89.63	89.53
Oracle combination	90.59	90.66	90.62
Improvement over baseline	+1.15	+1.03	**+1.09**
	Section 23 (testset)		
Baseline	88.77	88.87	88.82
Oracle combination	90.11	90.11	90.11
Improvement over baseline	+1.34	+1.24	**+1.29**

Table 1: Results of the pilot experiment

Table 1 demonstrates that the right combination of the parsers can yield a potential absolute performance increase

[1] http://www.speech.sri.com/projects/srilm/

[2] Note that we allowed duplicates in the original treebank before weighting.

[3] The harmonic mean of labeled precision (LP) and labeled recall (LR): F-score $= \frac{2*LP*LR}{LP+LR}$.

Individual Subdomain Parser	LR	LP	F-score
	Section 00 (devset)		
Sports	88.95	88.83	88.89
Financial	89.01	88.84	88.92
Politics	88.86	88.70	88.78

Table 2: Individual subdomain parser results

of 1.09% and 1.29% F-score compared to the baseline, accounting for an approximately 10% relative error reduction. The individual subdomain specific parsers, in contrast, are less accurate than the baseline when used each on their own (see table 2), which is expected since they are meant to be more specialized parsers.

Figure 2 exemplifies how subdomain sensitive parsing may improve over the original parser; it shows the relevant parts of the parse trees and the overall parsing performance for sentence #90 from the devset:

> *South Korea registered a trade deficit of $ 101 million in October, reflecting the country's economic sluggishness, according to government figures released Wednesday.*

The example illustrates that a domain-specifically trained parser may indeed find a correct or better sentence analysis than the baseline parser. The pilot study reveals that our domain-dependent parsing instantiation has potential. We next will see the results of our parser combination techniques that aim at achieving this potential.

Results of Parser Combination Techniques Table 3 reports the accuracy results for the selected parser combination techniques on the test as well as the development set for sentences up to 40 words.

Parser / Combination Technique	LR	LP	F-score
	Section 00 (devset)		
Baseline	89.44	89.63	89.53
Divergence Model (DVM)	89.50	89.68	89.59
Node Weight. incl. DVM, $\lambda = 0.6$	89.53	89.71	**89.62**
	Section 23 (test set)		
Baseline	88.77	88.87	88.82
Divergence Model (DVM)	88.80	88.91	88.85
Node Weight. incl. DVM, $\lambda = 0.6$	88.84	88.96	**88.90**

Table 3: Results of Parser Combination Techniques

Both the parser pre-selection and the parser post-selection technique can improve the baseline system slightly. The best technique is Node Weighting including the Divergence Model (NW-DVM), yielding an improvement of +0.09% and +0.08%. These figures are only about one tenth of the +1.29% that an oracle obtains (see table 1).

In figure 3 we depict the performance of NW-DVM when instantiated with various values for λ. We can see that for $\lambda \geq 0.5$ the technique slightly but constantly outperforms the baseline system, reaching a peak at $\lambda = 0.6$.

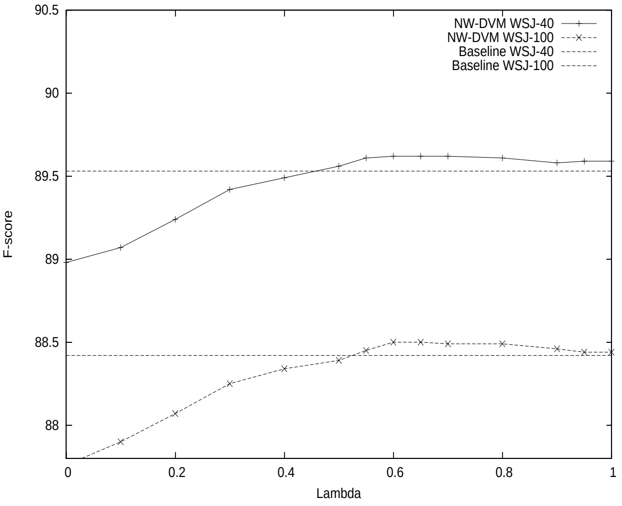

Figure 3: Result of NW-DVM for sentences up to 40 (WSJ-40) and 100 (WSJ-100) words, respectively.

4. Related Work

In (Bod, 1999), dialogue context/state is taken into account by **splitting** up the training treebank into four disjoint context-dependent subcorpora (place, time, date, yes/no). The "subdomain-dependent" parsers yield promising results. For a tiny domain as the one explored by Bod (train information system) this simplistic approach could work given that the treebank is large enough. However, in the general case, splitting the training data into disjoint subdomains implies sparse data problems for each of the subdomain parsers.

A further related study is the work by (Sekine, 1997). He analyzes the "domain dependence of parsing". In his experiments a domain is characterized by the natural domains defined in the Brown corpus, for example 'Press Reportage', 'General Fiction' or 'Romance and Love Story'. Sekine observes that in parsing, the data from the same domain is the most advantageous, followed by data from the same class, while training on data from another domain generally performs worst. Sekine claims that when trying "to parse a text in a particular domain, we should prepare a grammar which suits this domain" (Sekine, 1997), thus suggesting a "domain-dependent parser".

Although different, work on domain adaptation is distantly related to our work. Recent research on adaptation is too numerous to discuss in a short paper. In particular, (Jiang and Zhai, 2007) suggest "instance weighting" as a method for adaptation. Our approach, subdomain instance weighting using raw data, can be seen as a novel version thereof. Theoretically speaking, successfull domain adaptation hinges on some sense of "overlap" between the source and target domains, e.g., (Ando and Zhang, 2005). The overlap between source and target domains can be seen as a (mix of) subdomain(s) of both. Naturally, instance weighting and its subdomain instantiation can be seen as a weighted version of self-training, e.g., (McClosky et al., 2006), which is again related to co-training (Blum and Mitchell, 1998).

5. Conclusions and Outlook

This paper explores a particular instantiation for subdomain sensitive parsing. We exploit unlabeled subdomain corpora gathered from the web in order to create a set of subdomain

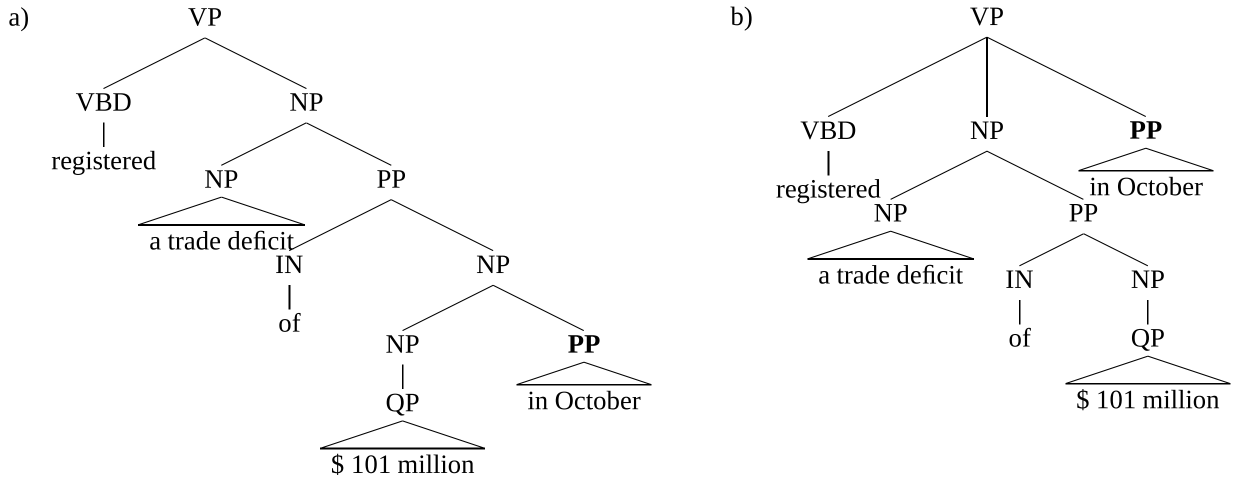

Figure 2: Part of the parse tree for sentence #90 (a) predicted by the baseline parser (F-score of tree: 87.80%; incorrect PP-attachment), and (b) corresponding oracle prediction: in this case, either of $Parser_{\text{FINANCIAL}}$ or $Parser_{\text{POLITICS}}$ (F-score of tree: 100%; correct PP-attachment).

sensitive parsers, where each parser is assumed to specialize in its domain.

The empirical results demonstrate that our subdomain Tree Weighting method is promising as it yields an improvement of up to 1.29% F-score on the Penn Treebank Wall Street Journal. However, the experiments also show that parser combination is not a straightforward task: amalgamating parser predictions gives only a modest improvement (+0.09%) over the (very tough) baseline system.

Given the simplicity of the Treebank Weighting scheme and the examined parse(r) selection methods, the experimental results are promising and warrant further exploration. Therefore, future work will be manifold and consists in, amongst others: defining other ways of instantiating subdomain sensitive parsers by using a more sophisticated notion of subdomain (e.g. integrating relations between entities), extending the current approach to a Bayesian approach, exploring more sophisticated parser combination techniques, and examining the approach on corpora other than the WSJ, e.g., Brown corpus (see (Gildea, 2001)), thus looking at domain adaptation. Furthermore, we would like examine our approach with the current best-performing parsing systems (Charniak and Johnson, 2005). It consists of a generative n-best parsing system that employs a discriminative reranking scheme. We would like to gauge to what extent n-best parsing might benefit from subdomain information.

6. References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Bikel, D. (2002). Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. Proceedings of HLT 2002.

Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100.

Bod, R. (1999). Context-sensitive spoken dialogue processing with the DOP model. In *Natural Language Engineering*. Cambridge University Press.

Charniak, E. and Johnson, M. (2005). Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005).

Collins, M. (1997). Three Generative, Lexicalised Models for Statistical Parsing. Madrid. Proceedings of the 35th Annual Meeting of the ACL-EACL.

Collins, M. (2003). Head-Driven Statistical Models for Natural Language Parsing. Association of Computational Linguistics.

Gildea, D. (2001). Corpus variation and parser performance. In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of ACL 2007*, pages 264–271, Prague, Czech Republic, June. Association for Computational Linguistics.

Kneser, R. and Peters, J. (1997). Semantic clustering for adaptive language modeling. volume 02, page 779, Los Alamitos, CA, USA. IEEE Computer Society.

Koehn, P. (2005). Europarl: A Multilingual Corpus for Evaluation of Machine Translation. MT Summit.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 13.

D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the COLING-ACL 2006*. The Association for Computer Linguistics.

Wikimedia Foundation Inc. (2007). Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/.

Sekine, S. (1997). The Domain Dependence of Parsing. Washington, DC, USA. The Fifth Conference on Applied Natural Language Processing.