Fast Domain Adaptation for Part of Speech Tagging for Dialogues

Sandra Kübler, Eric Baucom
Indiana University
{skuebler,eabaucom}@indiana.edu

Abstract

Part of speech tagging accuracy deteriorates severely when a tagger is used out of domain. We investigate a fast method for domain adaptation, which provides additional in-domain training data from an unannotated data set by applying POS taggers with different biases to the unannotated data set and then choosing the set of sentences on which the taggers agree. We show that we improve the accuracy of a trigram tagger, TnT, from 85.77% to 86.10%. In order to improve performance on unknown words, we investigate using active learning for learning ambiguity classes of domain specific words, yielding an accuracy of 89.15% for TnT.

1 Introduction

Part of speech (POS) tagging for English is often considered a solved problem. There are well established approaches such as Markov model trigram taggers (Brants, 2000), maximum entropy taggers (Ratnaparkhi, 1996), or Support Vector Machine based taggers (Giménez and Màrquez, 2004), and accuracy reaches approximately 97%.

However, most experiments in POS tagging for English have concentrated on data from the Penn Treebank (Marcus et al., 1993). If POS taggers trained on the Penn Treebank are used to tag data from other domains, accuracy deteriorates significantly. Blitzer et al. (2006) apply structural correspondence learning for learning pivot features to increase accuracy in the target domain. However, their approach is restricted to discriminative approaches to POS tagging.

In this paper, we investigate a simple and fast method for domain adaptation that is usable with any POS tagger: selecting reliably tagged in-domain data to add to the training set. This method has been successful for domain adaptation for dependency parsing (Chen et al., 2008). We use a corpus of dialogues collected in a collaborative task as target domain, thus introducing the challenges of processing spontaneous speech: hesitations, corrections, false starts, and contractions. We assume that this domain is more challenging than a target domain of biomedical texts, which is often used for domain adaptation research. Spontaneous speech dialogues do not only differ in terminology, but also in the types of sentences. Dialogues, for example, contain a higher percentage of questions and imperatives than formal written language, such as news or scientific writings.

Our domain adaptation experiments concentrate on adding in-domain training data based on an ensemble of POS taggers. The experiments show that extending the training set generally improves POS tagging accuracy. However, it cannot provide information on the ambiguity classes for words that do not appear in the source domain. For this reason, we integrate an active learning strategy for adding ambiguity classes for words that are identified automatically as unlikely to be tagged correctly.

The remainder of the paper is structured as follows: In section 2, we provide an overview of domain adaptation techniques in POS tagging and parsing. Section 3 describes our approach to domain adaptation, and section 4 describes the experimental setup. In section 5, we discuss our findings for domain adaptation, and in section 6, we describe the active learning extension.

2 Related Work

Domain adaptation is a task that has received much attention in recent years, with different results, ranging from evaluations that it is "frustratingly easy" (Daume III, 2007) to "frustratingly hard" (Dredze et al., 2007). The main differentiating factor seems to be whether a small portion

of annotated in-domain training data is available or only a large-size, unannotated data set. In our work, we concentrate on the second, more difficult, scenario.

Most work on domain adaptation has focused on parsing rather than on POS tagging (e.g. (McClosky et al., 2006; Yoshida et al., 2007; Chen et al., 2008; Rimell and Clark, 2008). Chen et al. (2008)) perform domain adaptation for a dependency parser. They show their best results are reached by adding only a selection of the information provided by a parser trained on out-of-domain data. Since short dependencies are more reliable than long ones, they select only the short, and thus reliable, ones and gain an increase in accuracy. Rimell and Clark (2008) adapt the Penn Treebank to parse grammatical relations in the biomedical domain. They report that the domains are similar structurally and that the lexicon is the main difference between the domains. Yoshida et al. (2007) investigate the influence of an external POS tagger on parsing accuracy in an HPSG parser. They show that the quality of the POS tagger has a significant influence even in-domain. The situation can be improved by allowing the POS tagger to output multiple, weighted POS tags from which the parser can choose. They show that allowing the tagger to output multiple POS tags improves parsing results both in-domain and out-of-domain.

Clark et al. (2003) use the results of one POS tagger on unannotated data to inform the training of another tagger in a semi-supervised setting using a co-training routine with a Markov model tagger and a maximum entropy tagger. The authors test both agreement-based co-training, where the sentences are added to training only if the taggers both agree, and naive co-training, where all sentences from one tagger are added to the training of the other, with no filter. For small sets of seed sentences, both types of co-training improve accuracy, with the higher quality, smaller training set from agreement-based co-training performing slightly better. The authors also report results for using naive co-training after the taggers were already trained on large amounts of manually annotated data. Naive co-training did not improve the taggers when trained in such a way (the authors leave agreement based co-training to future work).

Blitzer et al. (2006) investigate domain adaptation for POS tagging using the method of structural correspondence learning (SCL). SCL provides an informative feature-space for modeling the similarities between source and target domain by identifying *pivot* features. Pivot features behave similarly across domains, and if non-pivot features in the different domains correspond to many of the same pivot features, they are assumed to correlate. The machine learning algorithm is trained with the feature-space model from SCL on the source domain, with the idea that the trained model will now be informative for the unlabeled target domain as well. Blitzer et al. (2006) evaluate the SCL transfer of a POS tagger from the Penn Treebank to a corpus of biomedical abstracts (MEDLINE), reporting an improvement from 87.9% to 88.9%. The authors report that vocabulary is the main difference between the domains. However, SCL can only be applied to feature-based discriminative learning methods.

3 Domain Adaptation by Tagger Combination

For our experiments, we use the Wall Street Journal part of the Penn Treebank as source domain and dialogues in a collaborative task as target domain. In the target domain, we have access to a large unannotated corpus and a small annotated corpus, which we use for evaluation purposes. In order to adapt a POS tagger to the target domain, we extend the training set by sentences from the large unannotated corpus. Our hypothesis is that these sentences will provide the POS tagger with relevant information from the target domain. For assigning POS tags to the additional sentences from the target domain, we use three different POS taggers trained on the Penn Treebank. Then we select those sentences for which a majority of taggers agree on the POS tags. The method of using agreement between taggers was originally used by van Halteren et al. (2001) to improve tagger performance. We investigate the following questions: 1) How does the number of agreeing POS taggers influence the accuracy of the final tagger? 2) Should we select only complete sentences or add all trigrams on which the taggers agree? Lifting the restriction that the taggers agree on complete sentences will increase the size of the training set. 3) Do we need the full Penn Treebank training set, or does this large training set dominate the smaller training set from the target domain?

4 Experimental Setup

4.1 Data Sets

We use three corpora: the Penn Treebank for the source domain; the HCRC Map Task Corpus (Thompson et al., 1996) for additional training in the cooperative dialogue domain; and the CReST corpus (Eberhard et al., 2010) for evaluation in the target domain.

The HCRC Map Task Corpus (Thompson et al., 1996) is a multi-modal corpus composed of 18 hours of digital audio and 150 000 words of transcription, representing 128 two-person conversations. The conversations were obtained from a cooperative problem solving task, in which two participants were asked to help one another fill in a route on a map. HCRC is annotated for speaker and dialogue turn information, as well as for POS tags. However, we use only the actual transcriptions. This corpus serves as our unannotated, in-domain training corpus.

The CReST Corpus (Eberhard et al., 2010) is a multi-modal corpus consisting of 7 dialogues, comprising 11 317 words in 1 977 sentences. Similar in domain to the HCRC corpus, it represents cooperative dialogues, but is based on a slightly different task: one of the participants is located in a search environment, while the other is outside but has access to a map of the environment. The participants need to collaborate to fulfill their tasks (locating objects in the environment and placing objects on the map).

CReST is annotated for POS, syntactic dependency and constituency, disfluency, and dialogue structure. The POS tagset is a superset of the tagset for the Penn Treebank, with the additional tags representing features unique to natural dialogue.

Data Preparation. Due to differences between the transcriptions of HCRC and CReST, we made small, systematic changes to HCRC to make it more consistent with CReST. For instance, HCRC had various permutations of `mmhmm` which we changed to the standard `mhm` transcription in CReST. Since the Penn Treebank does not contain all tags used in CReST, we translated the additional CReST tags into tags of the original tagset for our experiments. E.g. the POS tag `VBI` (imperative verb) is translated into `VB` (verb in the base form).

4.2 POS Taggers

We use three POS taggers: TnT (Brants, 2000), MElt (Denis and Sagot, 2009), and SVMTool (Giménez and Màrquez, 2004). These taggers were chosen because they represent the state of the art for single-direction taggers and also because they use different approaches to POS tagging and thus have different biases. Our assumption is that the different biases will result in different types of POS tagging mistakes.

TnT (Brants, 2000) is a trigram Markov model POS tagger with state-of-the-art treatment of unknown words. TnT generates files containing lexical and transition frequencies and thus provides us with the option of including new trigrams directly into the trained model.

The Maximum-Entropy Lexicon-Enriched Tagger (MElt) (Denis and Sagot, 2009) is a conditional sequence maximum entropy POS tagger that uses a set of lexical and context features, which are a superset of the features used by Ratnaparkhi (1996) and Toutanova and Manning (2000).

SVMTool (Giménez and Màrquez, 2004) is a discriminative POS tagger based on support vector machines. The features and specifications used in training were taken from the SVMTool model for English, based on the Penn Treebank.

5 Experiments

We perform six experiments: The first experiment establishes a baseline by training the POS taggers out of domain on the Penn Treebank and then using them without adaptation on the target domain. In the second experiment, the training set is extended by those HCRC sentences on which all three taggers agree. In the third experiment, we investigate whether the accuracy of the adapted tagger deteriorates if we choose all sentences on which only two taggers agree. In the fourth experiment, we investigate the effect of adding trigram information on which all taggers agree to the TnT trained model. In the fifth experiment, we also add lexical information to the TnT model. In the final experiment, we investigate whether the large size of the Penn Treebank neutralizes effects from the additional training data, based on the experiment with sentences on which all three taggers agree.

Tagger	baseline	all3
MElt	83.91	84.32†
SVMTool	84.60	85.15†
TnT	**85.77**	85.70

Table 1: The results of the baseline and of selecting all sentences on which all taggers agree. Dags indicate a significant improvement over the baseline.

5.1 Agreement Among All POS Taggers

This experiment uses all three POS taggers, trained on the Penn Treebank, to tag all sentences from the HCRC corpus. Then all sentences are selected on which the taggers agree. These sentences are added to the Penn Treebank training set, and the taggers are retrained and evaluated on the CReST corpus. The results of the baseline and this experiment are shown in table 1.

The results show that both discriminative POS taggers, MElt and SVMTool, improve significantly over the baseline (McNemar, $p < 0.001$). TnT, in contrast, suffers a non-significant decrease in performance. However, TnT's baseline results are significantly higher than the two other taggers'. This can be explained by the state-of-the art module for guessing unknown words in TnT, which is based on suffix tries extracted from hapax legomena in the training data set. For the baseline, TnT reaches an accuracy of 16.64% on unknown words, MElt 11.65%, and SVMTool 10.32%.

In order to determine whether our initial low performance was due to within-domain tagging issues, such as "fuzzy" linguistic boundaries (Manning, 2011), or simply to the level of difference between our source and target domains, we conducted a brief analysis of the errors from this experiment. We found that the top three discrepancies in the `all3` condition for TnT, comprising 55.32% of the incorrect tags, were the result of mistakenly labeling a gold-tagged interjection (`UH`) with an adjective (`JJ`), noun (`NN`), or an adverb (`RB`) tag. The next most common mistake was labeling a gold-tagged `SYM` (incomplete or non-word) with `JJ` (5.32% of discrepancies). `SYM` and `UH` are much more common in a corpus of spoken dialogue transcriptions than in closely edited financial news. Thus, these top four mistakes represent errors arising from the dissimilarity of the domains (as opposed to the fifth mistake, mistaking `IN` (preposition) for `RB`, which is a more traditional within-domain tagging error, with "fuzzy" linguistic boundaries partially to blame).

Training	# of words
baseline	1 342 561
all3	1 391 238
me/svm	1 413 106
me/tnt	1 418 957
svm/tnt	1 412 917

Table 2: Number of words in the training set.

Tagger	me/svm	me/tnt	svm/tnt	all3
MElt	84.37†	84.28	84.59†	84.32†
SVM	84.98	85.30†	85.47†	85.15†
TnT	**85.94**	85.84	85.70	85.70

Table 3: Results of adding all sentences for which two taggers agree.

5.2 Agreement Between Two Taggers

The reason for requiring all three POS taggers to agree on full sentences is that the selected sentences will be reliable. However, the method also has the drawback that only a rather small number of sentences fulfill this criterion. The first 2 rows in table 2 show the number of words in the training data for the baseline experiment with only Penn Treebank data and for the `all3` experiment. They show that only a very small number of words is added: The number of words increases from approximately 1.34 million words to 1.39 million, i.e. only 50 000 words are added out of the 150 000 words in the HCRC corpus, an insignificant number when compared to the source domain data.

Thus, in order to provide more in-domain training data, we relax the constraint on the selection of sentences from the HCRC corpus and select all sentences for which two specific taggers agree. The results are shown in table 3. The last column in this table repeats the results from the previous experiment.

These results show that the additional data (cf. table 2) improves performance over the experiment requiring agreement between all three taggers. It is worth noting that MElt and TnT perform best with training where the common sentences are from the two *other* taggers. For SVMTool, including TnT improves accuracy, but there is no significant difference between the combination of MElt with TnT and the one with SVMTool

and TnT. We assume that TnT has reached saturation on the Penn Treebank and cannot learn new information from additional data tagged with its own bias. Sentences from the other taggers, however, do present new information.

We had a closer look at the sentences that were added to TnT when MElt and SVMTool (me/svm) agree and when MElt and TnT (me/tnt) agree and found considerable differences in the distribution of POS tags. These differences can also be found in the test set tagged with TnT, based on the two training sets. In all data sets, the combination me/tnt seems to keep the lexical bias of the Penn Treebank more strongly than the combination me/svm. For example, the word `left` is consistently tagged as a noun when TnT uses the me/svm combination. In most instances, this is the correct decision. The me/tnt combination, in contrast, prefers a verb reading. For the word `back`, the me/svm combination selects the correct adverb reading over the verbal particle reading preferred by the me/tnt reading. Since the combination of SVMTool and TnT also keeps the bias, the innovation in the me/svm combination cannot be attributed to having SVMTool in the combination.

We also investigated whether using a union of sentences from different pairs of taggers would increase overall accuracy. This adds approximately 70 000 words to the training set. However, the results of this experiment proved to be not significantly different from those based on tagger pairs.

5.3 From Complete Sentences to n-grams

The results from the previous experiment show that adding more training data, even if it is less certain, improves the accuracy of the final tagger. One possibility to provide more training material consists in relaxing the constraint that the taggers need to agree on complete sentences. Instead, we extract either all longest matching n-grams or all trigrams on which the taggers agree. The n-grams are processed and added to the TnT model from the Penn Treebank. This is only possible because TnT stores its trained model in an accessible format. The discriminative POS taggers could not be used for this experiment since adding incomplete sentences as training data would have influenced their trained models negatively.

As before, all evaluations are performed on CReST. The results of this experiment are shown in table 4. The first 3 columns contain the results of merging n-grams or trigrams from 2 different taggers; the last column shows the results for merging all 3 taggers. The first row repeats the results from previous experiments using complete sentences that taggers agree upon. We restrict ourselves to adding only transition information here and merely use the lexicon from the Penn Treebank baseline. We will investigate adding both transition and lexical information in the next experiment. The results show that adding trigrams instead of complete sentences, based on MElt and SVMTool, results in approximately 25 000 additional trigram counts, and it improves the accuracy of the final tagger from 85.94% to 86.10%. Adding all n-grams, in contrast, adds around 33 000 trigram counts and results in slightly lower accuracies, demonstrating that in some cases, the sheer amount of data may be counteracted by substandard quality. Again, TnT profits most from in-domain sentences provided by a combination of MElt and SVMTool.

	me/svm	me/tnt	svm/tnt	all3
full	85.94	85.84	85.70	85.70
n	85.88	85.55	85.93	85.76
tri.	**86.10**	85.77	85.93	85.98

Table 4: Results of adding n-grams or trigrams to TnT's model.

5.4 Adding Lexical Information

A look at the words that are mistagged with the highest frequency in the previous experiment, in which we added trigram information, shows that they fall into two different categories: words such as `okay`, `um`, `gonna` that are typical for dialogues but do not occur frequently in the Penn Treebank; and words that have a different POS preference in the target domain. An example for this category is the word `left`, which tends to be a verb in the Penn Treebank and an adverb in CReST.

For this reason, we decided to add the lexical information from the trigrams to TnT's lexicon. The results of this experiment are shown in table 5. They show that adding lexical information results in lower accuracies: they decrease minimally from 86.10% (adding only trigram transition information) to 86.00% when adding both transition and lexical information. When adding n-grams and lexical information, the results improve over adding only n-grams, but they do not reach the

	me/svm	me/tnt	svm/tnt	all3
n	85.88	85.55	85.70	85.70
n+lex.	86.00	85.86	85.81	85.88
tri.	**86.10**	85.77	85.93	85.98
tri.+lex.	86.00	85.42	85.78	85.86

Table 5: Results of adding lexical information to TnT's model.

best trigram result.

Since this result did not meet our expectation, we analyzed the changes to the lexicon file and the tagging errors. The extended lexicon contains 326 additional words, but only 8 of them also occur in CReST (`ah`, `fifteen`, `forty-five`, `furthest`, `hmm`, `mhm`, `um`, `yeah`). `yeah` is by far the most common word in the test data. The small number of added words that actually occur in the test set severely restricts possible improvements on in-domain POS tagging.

A comparison of TnT's performance with and without the extended lexicon shows that there are 101 discrepancies (in 11 317 words) in which the POS tagger without additional lexical information makes the correct decision. Out of these discrepancies, the word `yeah` accounts for 45 errors. Here, the extended lexicon lists the tag `NN` instead of the tag `UH`. The reason lies in the fact that `yeah` does not occur in the Penn Treebank, and the three taggers trained on this treebank all (wrongly) tag `yeah` with the most frequent tag for unknown words.

From the error analysis, we can conclude that the added words do not correspond to the words that are needed in the test domain, which means that the HCRC map task corpus data are not similar enough to the CReST data. However, we can also conclude that even if there were a larger overlap, there is a high chance that those words would be mistagged by the ensemble of taggers so that adding the new words would result in a deterioration of the performance.

5.5 Decreasing Out-Of-Domain Training Data

In a final experiment, we investigate whether the difference in amounts of training data between source domain and target domain neutralizes the positive influence of adding in-domain information. Table 2 shows that the number of words added by our methods ranges between one third and half of the original data set. It is therefore possible that the added information does not change the transition probabilities enough to improve the behavior of the final tagger. For this reason, we restrict the size of the Penn Treebank training set to the number of words in the in-domain set, thus reducing the influence of the out-of-domain data. For this experiment, the in-domain training set is taken from the combination of all 3 taggers, as reported in table 1. The results of this experiment are shown in table 6.

Tagger	baseline	red. base.	red.+all3
MElt	83.91	79.38	83.86
SVMTool	84.60	78.79	83.90
TnT	85.77	79.86	84.11

Table 6: The results of restricting the size of the out-of-domain training set.

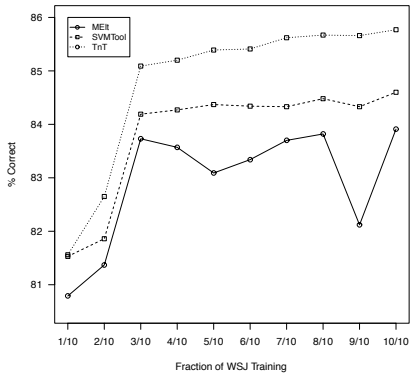

Figure 1: Accuracy as a function of amount of WSJ training

The results show that training the POS tagger on only the reduced Penn Treebank containing 46 680 words results in a severe loss in accuracy, from approximately 84% to approximately 79%. Adding more training data from the Penn Treebank consistently increases the results, as shown in Figure 1, thus demonstrating that more data is more important than in-domain knowledge.

These experiments shows that even a fairly "easy" problem such as POS tagging requires a large training set. In the first experiment, combining the reduced Penn Treebank with the in-domain data set increases the accuracies of all POS taggers over the reduced baseline, but they do not reach the baseline based on the whole Penn Treebank. This experiment shows that the sheer size of the training set is more important than access to in-domain training data, at least when the quality of

	base	all3	me/svm	me/tnt	svm/tnt
trans.	85.77	86.10	85.77	85.93	85.98
active	89.04	89.13	**89.15**	89.03	**89.15**

Table 7: Results of adding an active learning lexicon to the training for TnT. All differences between the two experiments are significant.

this additional training set is not guaranteed.

6 Extending the Lexicon with Active Learning

The results of the previous sections show that adding information on which taggers trained out-of-domain agree is useful for moderately improving tagging accuracy and especially for reestimating transition probabilites. However, the method is unsuitable for finding the correct ambiguity sets for words that do not occur in the out-of-domain training set. Such words must be treated in the tagger's module for handling unknown words, which is often based on suffix information extracted from infrequent words in the training set. However, many of the unknown words in the CReST corpus are colloquial words and thus do not show the same morphological characteristics as words in the training set. The words `yeah` and `mhm` are good examples: it is rather unlikely that the tagger can guess their ambiguity set based on their bigram suffixes `ah` and `hm`. This problem is not unique to the domain of spontaneous speech. Biomedical terms, for example, also display atypical suffixes, which make them difficult to classify.

Since the training corpus cannot provide the required information, we decided to acquire minimal information from the target domain via active learning. This goal here is to automatically identify words that TnT was likely to tag incorrectly. These words are then presented to the user, who is asked to provide the ambiguity sets for the words. In our experiment, we simulated the user by looking up the words identified by our program in the CReST gold standard.

In order to determine which words would be difficult for TnT, we built a suffix trie similar to TnT's model for unknown words. For the sake of simplicity, we restricted the trie to a maximum suffix length of three letters. Then, each word in our CReST test corpus that did not occur in the Penn Treebank training lexicon was matched against the suffix trie. If the word's suffix was not present in the trie, the word was presented to the user and added to TnT's lexicon. The extended lexicon was used in combination with the extended transitions based on trigrams from section 5.3. In total, 74 ambiguity classes were added in the active learning lexicon.

The results in table 7 show that adding the active learning lexicon to the Penn Treebank baseline improves tagging accuracy to 89.04%, outperforming our best previous results (cf. table 4). The best results of 89.15% are based on combinations of the active learning lexicon and transition information from where just two taggers agree on HCRC trigrams. This illustrates that adding new words to the lexicon results in a higher improvement than adding new transition information. However, the best results are gained by a combination of the two methods. All active learning results are significantly higher than the previous best result of 86.10%.

For the Penn Treebank baseline, there were 176 word types that were wrongly tagged. In the active learning experiment, 71 types (40.34%) were added with their ambiguity classes, among them the prevalent word `yeah`. All of these words were unambiguous in the target domain.

7 Conclusion and Future Work

We investigated a generally applicable method of domain adaptation for POS tagging, which uses the consent of three POS taggers with different biases to add in-domain sentences to the training set. We show that we reach a slight but significant increase in accuracy from 85.77% to 86.10% when using all trigrams on which the POS taggers agree. Reducing the size of the out-of-domain training set has a detrimental effect on the quality of the POS tagger. The improvement from adding in-domain trigrams is due to more accurate transition probabilities. In contrast, the lexical additions from the in-domain data were detrimental. The active learning strategy of adding user-defined lexical information for difficult unknown words improves this accuracy to 89.15%. However, this accuracy is still far below an in-domain accuracy, which

reaches 95.66%.

TnT's better performance on this task may be due to its superior handling of unknown words, but may also be a result of the fact that the feature sets used with MElt and SVMTool were designed specifically for the Penn Treebank. We may be able to improve results for those two taggers if we optimize the feature set for the target domain. However, this means modifying the implementation of the taggers since the feature extraction is not modular. For the future, we are planning to investigate whether structural correspondence learning (Blitzer et al., 2006) will reach higher accuracies, even though it cannot be used with our best performing POS tagger, TnT. We will also repeat these experiments with a biomedical target domain to see if our results transcend domains.

Acknowledgment

This work is based on research supported by the US Office of Naval Research (ONR) Grant #N00014-10-1-0140.

References

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia.

Thorsten Brants. 2000. TnT–a statistical part-of-speech tagger. In *Proceedings of the ANLP-NAACL*, Seattle, WA.

Wenliang Chen, Youzheng Wu, and Hitoshi Isahara. 2008. Learning reliable information for dependency parsing adaptation. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, Manchester, UK.

Stephen Clark, James Curran, and Miles Osborne. 2003. Bootstrapping POS-taggers using unlabelled data. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL)*, Edmonton, Canada.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Pascal Denis and Benoit Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC 23)*, Hong Kong, China.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, Czech Republic.

Kathleen Eberhard, Hannele Nicholson, Sandra Kübler, Susan Gunderson, and Matthias Scheutz. 2010. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of LREC*, Lisbon, Portugal.

Christopher Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of CICLing*, Tokyo, Japan.

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of COLING-ACL*, Sydney, Australia.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP*, Philadelphia, PA.

Laura Rimell and Stephen Clark. 2008. Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of EMNLP*, Honolulu, Hawaii.

Henry Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. 1996. The HCRC Map Task Corpus: Natural dialogue for speech recognition. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP-VLC*, Hong Kong.

Hans van Halteren, 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.

Kazuhiro Yoshida, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical intelligence*, Hyderabad, India.