Exploring Options for Fast Domain Adaptation of Dependency Parsers

Viktor Pekar, Juntao Yu, Mohab El-karef, Bernd Bohnet
School of Computer Science
University of Birmingham
Birmingham, UK
{v.pekar,jxy362,mxe346,b.bohnet}@cs.bham.ac.uk

Abstract

The paper explores different domain-independent techniques to adapt a dependency parser trained on a general-language corpus to parse web texts (online reviews, newsgroup posts, weblogs): co-training, word clusters, and a crowd-sourced dictionary. We examine the relative utility of these techniques as well as different ways to put them together to achieve maximum parsing accuracy. While we find that co-training and word clusters produce the most promising results, there is little additive improvement when combining the two techniques, which suggests that in the absence of large grammatical discrepancies between the training and test domains, they address largely the same problem, that of unknown vocabulary, with word clusters being a somewhat more effective solution for it. Our highest results were achieved by a combination of word clusters and co-training, significantly improving on the baseline, by up to 1.67%. Evaluation of the best configurations on the SANCL-2012 test data (Petrov and McDonald, 2012) showed that they outperform all the shared task submissions that used a single parser to parse test data, averaging the results across all the test sets.

1 Introduction

Domain adaptation of a statistical dependency parser is a problem that is of much importance for many practical NLP applications. Previous research has shown that the accuracy of parsing significantly drops when a general-language model is applied to narrow domains like financial news (Gildea, 2001), biomedical texts (Lease and Charniak, 2005), web data (Petrov and McDonald, 2012), or patents (Burga et al., 2013). In a preliminary experiment, we looked at the effect of cross-domain parsing on three state-of-the-art parsers – Malt (Nivre, 2009), MST (McDonald and Pereira, 2006), and Mate parser (Bohnet et al., 2013) – trained on the CoNLL09 dataset and tested on texts from different domains in the OntoNotes v5.0 corpus as well as the in-domain CoNLL09 test set. The results (see Table 1) indicate that depending on the application domain, the parsing accuracy can suffer an absolute drop of as much as 16%.

Domain	MST	MALT	Mate
Newswire	84.8	81.7	87.1
Pivot Texts	84.9	83.0	86.6
Broadcast News	79.4	78.1	81.2
Magazines	77.1	74.7	79.3
Broadcast Conversation	73.4	70.5	74.4
CoNLL09 test	86.9	84.7	90.1

Table 1: Labelled accuracy scores achieved by the MST, Malt, and Mate parsers trained on CoNLL09 data and tested on different specialist domains.

In a typical domain adaptation scenario, there are **in-domain texts** that are manually annotated and that are used to train a general-language parser, and **out-of-domain** or **target domain texts** that are

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: http://creativecommons.org/licenses/by/4.0/

First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages, pages 54–65,
Dublin, Ireland, August 23-29 2014.

parsed during parser testing. In addition, a certain amount of unlabelled target domain texts may be available that can be leveraged in this or that way to facilitate domain adaptation. To address the problem of domain adaption, previous work focused on weakly supervised methods to re-train parsers on automatically parsed out-of-domain texts, through techniques such as co-training (Sarkar, 2001; Steedman et al., 2003), self-training (McClosky and Charniak, 2008; Rehbein, 2011), and uptraining (Petrov et al., 2010); selecting or weighting sentences from annotated in-domain data that fit best with the target domain (Plank and Van Noord, 2011; Søgaard and Plank, 2012; Khan et al., 2013b). Another line of research aims specifically to overcome the lexical gap between the training data and the target domain texts. These approaches include techniques such as text pre-processing and normalization (Foster, 2010), the use of external lexica and morphological clues to predict PoS tags of unknown target domain words (Szolovits, 2003; Pyysalo et al., 2006), discrete or continuous word clusters computed from unlabelled target domain texts (Candito et al., 2011; Bansal et al., 2014), selectional preferences modelled from word co-occurrences obtained from unannotated texts (Zhou et al., 2011).

The goal of this paper is to investigate a combination of such techniques to adapt a general-language parser to parse web data (weblogs, online reviews, newsgroups, and answers) without resorting to manual annotation. In our study we include several techniques that have been shown to be reasonably effective for domain adaptation: text normalization, the use of word clusters, an external crowd-sourced lexicon, as well as automatically annotated texts produced with the help of co-training. All these techniques are domain-independent and can be applied to new target domains given unlabelled texts form these domains. We explore the relative utility of these methods and ways to combine them for maximum parser accuracy.

2 Related work

2.1 Text normalization

User-generated content on the web is notoriously low-quality, containing slang, abbreviations, inconsistent grammar and spelling. Foster (2010) investigated lexical phenomena that appear on online discussion forums that present common problems for parsing and compiled a list of such phenomena along with their transformations. Applying the transformations to test sentences helped to bring the F-score up by 2.7%. A similar approach was taken by Khan *et al.* (2013a) who found that it performed better than spelling correction based on the Levenshtein distance. Gadde *et al.* (2011) use a word clustering method and language modelling in order to align misspelled words with their regular spelling. Their method of cleaning noisy text helped to increase the accuracy of PoS tagging of SMS data by 3.5%.

2.2 External lexica

To adapt the Link parser to the medical domain, Szolovitz (2003) extended its lexicon with terms from the UMLS Specialist Lexicon. Pyysalo *et al.* (2006) take the same approach and together with predicting the PoS tags for out-of-vocabulary words based on their morphology this allowed them to achieve a 10% reduction in the error rate of parsing. External lexica have also been used to improve out-of-domain PoS tagging (Li et al., 2012).

2.3 Word clusters

In order to reduce the amount of annotated data to train a dependency parser, Koo *et al.* (2008) used word clusters computed from unlabelled data as features for training a parser. The same approach has proved to be effective for out-of-domain parsing, where there are many words in the test data unseen during training, and word clusters computed from in-domain data similarly help to deal with the vocabulary discrepancies between the training and test datasets. Discrete word clusters produced by Brown *et al.* (1992) method have been shown to be beneficial for adapting dependency parsers to biomedical texts (Candito et al., 2011) and web texts (Øvrelid and Skjærholt, 2012). Word clusters created with Brown clustering method have also been used to adapt a PoS tagger to Twitter posts (Owoputi et al., 2013). Bansal *et al.* (2014) introduced continuous word representations and showed them to increase parsing accuracy both on the Penn Treebank and on web data.

2.4 Co-training

Co-training (Blum and Mitchell, 1998) is a paradigm for weakly supervised learning of a classification problem from a limited amount of labelled data and a large amount of unlabelled data, whereby two or more views on the data, i.e. feature subsets, or two or more different learning algorithms are employed that complement each other to bootstrap additional training data from the unlabelled dataset. Co-training algorithms have been successfully used in NLP tasks, and specifically for parsing. Sarkar (2001) showed the both precision and recall of a phrase structure parser can be increased using a co-training procedure that iteratively adds the most confidently parsed sentences from two different views to the training set. Steedman *et al.*(2003) used two different parsers that supplied training data to each other in a bootstrapping manner.

A number of studies specifically aimed to use co-training for domain adaptation of a dependency parser. Sagae (2007) used two different learning algorithms of their graph-based parser to complete a one iteration of co-training, getting an improvement of 2-3%, which was the best result on the out-of-domain track of the CoNLL07 shared task (Nilsson et al., 2007). An interesting finding of their work was that the agreement between the two classifiers during testing was a very good predictor of accuracy. More recently, Zhang *et al.* (2012) used a tri-training algorithm for parser domain adaptation. The algorithm uses three learners and each learner was designed to learn from those automatically classified unlabelled data where the other two learners agreed on the classification label.

3 Experimental set-up

3.1 Parsers

In the experiments we included the Malt parser (Nivre, 2009), the MST parser (McDonald and Pereira, 2006), the transition-based Mate parser (Bohnet et al., 2013), and the graph-based Turbo parser (Martins et al., 2010). All the parsers were used with their default settings, and PoS tags used in the input of all the parsers were the same and came from the Mate parser.

3.2 Baseline

As the baseline we used the Mate parser, as it showed the highest accuracy when no domain adaptation techniques were used, i.e. trained on an in-domain training dataset and applied directly to out-of-domain test data.

3.3 Data

The experiments were conducted on annotated data on web-related domains available in the Ontonotes v.5 and SANCL datasets, since a large amount of unlabelled data required for most domain adaptation techniques is widely available.

OntoNotes. In experiments with weblog texts, we used the CoNLL09 training dataset (Hajič et al., 2009) as the general-language training data. The CoNLL09 test dataset was used to evaluate in-domain parsing. To create an out-of-domain test set, we selected the last 10% of the weblogs section of the OntoNotes v5.0 corpus[1], in order to make the size of the out-of-domain test data comparable to that of the in-domain test data, i.e. of CoNLL09 test. The OntoNotes corpus was converted to the CoNLL09 format using the LTH constituent-to-dependency conversion tool (Johansson and Nugues, 2007).

SANCL. In order to compare our results with the results achieved by participants in the SANCL-2012 shared task, we also ran experiments on the Stanford dependences of three SANCL test sets (`answers`, `newsgroups` and `reviews`). In these experiments we used the training set, test sets, unlabelled data, as well as the evaluation script provided by SANCL-2012 organizers (Petrov and McDonald, 2012).

Tables 2 and 3 show the sizes of the OntoNotes and SANCL datasets as well as several measures of lexical and grammatical characteristics of the data. The average sentence length (in tokens) and the average number of subjects, roughly corresponding to the number of clauses in the sentence, aim to characterize the syntactic complexity of the sentences: the higher these values, the more complex the

[1] https://catalog.ldc.upenn.edu/LDC2013T19

structure of the sentences is likely to be. The ratio of word forms absent from training data describes how different the train and test data are in terms of vocabulary.

We see that in the OntoNotes test set the average sentence length and the number of subjects per sentence is very similar to those in the train data. In SANCL test sets, these measures are more different, but the values indicate a smaller syntactic complexity than in the train data. The amount of unknown vocabulary in all the four test sets is between 5% and 8%.

	CoNLL09 train	**CoNLL09 test**	**OntoNotes test**
Sentences	39,279	2,399	2,150
Tokens	958,167	57,676	42,144
Sentence length	24.61	24.59	23.4
Subjects	1.8	1.83	1.89
Unk. wordforms ratio	0.0	0.011	0.05

Table 2: The size of OntoNotes train and test datasets.

	SANCL train	**Answers test**	**Newsgroups test**	**Reviews test**
Sentences	30,060	1,744	1,195	1,906
Tokens	731,678	28,823	20,651	28,086
Sentence length	24.56	18.44	22.79	16.35
Subjects	1.69	1.78	1.62	1.5
Unk. wordforms ratio	0.0	0.064	0.084	0.051

Table 3: The size of SANCL train and test datasets.

Unlabelled Data. As unlabelled target domain data we used the unlabelled dataset from the SANCL-2012 shared task. In experiments with word clusters, the entire dataset was used without any pre-processing. In the co-training experiments, we pre-processed the data by removing sentences that are longer than 500 tokens, or contained non-English words (this reduced the test set by 2%). Table 4 describes the size of the subsets of the unlabelled data.

	Emails	**Weblogs**	**Answers**	**Newsgroups**	**Reviews**
Sentences	1,194,173	524,834	27,274	1,000,000	1,965,350
Tokens	17,047,731	10,356,284	424,299	18,424,657	29,289,169

Table 4: The size of unlabelled datasets.

3.4 Evaluation method

As a measure of parser accuracy, we report labeled attachment scores (LAS), the percentage of dependencies which are attached and labeled correctly. Significance testing was performed using paired t-test.

4 Results and Discussion

4.1 Text normalization

We used a manually compiled lexicon containing Internet-specific spellings of certain words aligned with their traditional spellings, e.g. *u* ⇒ *you*, *gr8* ⇒ *great*, *don,t* ⇒ *don‘t*, as well as a number of regular expressions to deal with extra symbols usually added for emphasis (*This is sooooo good.*, *This *is* great.*). After the original word forms were read by the parser, the lexicon and the regular expressions were applied to normalize the spelling of the words. This produced only a very insignificant gain on the baseline. A manual examination of the test data in both OntoNotes and SANCL has shown that in fact although it comes from the web it contains very few examples of “Internet speak”.

4.2 Word clusters

We used Liang's (2005) implementation of the Brown clustering algorithm to create clusters of words found in unlabelled domain texts. The output of the algorithm are word types assigned to discrete hierarchical clusters, with clusters assigned ids in the form of bit strings of varying length corresponding to clusters of different granularity. We experimentally set the maximum length of the bit string to 6, collapsing more fine-grained clusters. Instead of replacing the original word forms and/or PoS tags with cluster ids as was done in some previous studies (Koo et al., 2008; Candito et al., 2011; Täckström et al., 2013), the ids of clusters were used to generate additional features in the representations of the word forms, as this also produced better results in the preliminary runs. Below we describe experiments with several other parameters of the clustering algorithm.

Number of clusters. As an input parameter, the Brown clustering algorithm requires a desired number of clusters. Initially discarding all word types with a count of less than 3, we experimented with different numbers of clusters and found that an optimal settings lies around 600 and 800 clusters, which gives an improvement on the baseline of 0.9% for out-of-domain texts; but there does not seem to be noticeable differences between specific numbers of clusters (see Table 5, statistically significant differences to the baseline are indicated by stars[2]).

Number of clusters	CoNLL09	OntoNotes
50	90.46**	78.10*
100	90.28*	78.40**
200	90.27	78.39**
400	90.37**	78.20**
600	90.40**	78.43**
800	90.30*	78.14**
Baseline	90.07	77.54

Table 5: The effect of the number of word clusters on in- and out-of-domain parsing, using the reviews and weblogs subsets of the SANCL-2012 unlabelled data.

Filtering rare words. Due to the inevitable data sparseness, the algorithm is likely to mis-cluster infrequent words. At the same time, it is rare words that are not seen during parser training and are potentially of greatest value if included into word clusters. We examined several thresholds on word frequency and their impact on parsing accuracy (see Table 6; statistically significant differences to the baseline are indicated by stars). We found very slight differences between these three thresholds, although the cut-off point of 3 showed the best results. Hence in further experiments with word clusters we used this cut-off point.

Min. freq.	CoNLL09	OntoNotes
1	90.36**	78.12*
3	**90.40****	**78.43****
5	90.22	78.24**

Table 6: The effect of filtering out rare words on word clusters, using the reviews and weblogs subsets of the SANCL-2012 unlabelled data.

Amount of unlabelled data. To examine the effect that the size of unlabelled data from which word clusters are computed, has on parser accuracy, we compared parser accuracy achieved when using only the reviews and weblogs subsets of the SANCL corpus (39.6 mln word tokens), and when using the entire SANCL dataset (75.2 mln tokens). These results are shown in Table 7, significant improvements on the smaller set are indicated by stars. As expected, a larger amount of data does improve the parsing accuracy, and the improvement is greater for out-of-domain parsing (+0.55% vs. +0.32%).

[2] In this and the following tables, one star indicates significance at the $p < 0.05$ level, two stars at the $p < 0.01$ level.

	CoNLL09	**OntoNotes**
Reviews and Weblogs	90.30	78.14
Entire SANCL dataset	**90.62***	**78.69***

Table 7: The effect of the size of unlabelled data on word clusters, discarding word types with count less than 3.

Relevant domain data. Furthermore, we were interested if simply adding more unlabelled data, not necessarily from the relevant domain, produced the same increase in accuracy. We obtained the plain-text claims and description parts of 13,600 patents freely available in the Global IP Database which is based on the Espacenet[3], creating a corpus with 42.5 mln tokens, i.e. which was similar in size to the reviews and weblogs sections of the SANCL unlabelled dataset. Table 8 compares results achieved when building clusters from the patents corpus and when using the reviews and weblogs texts from the SANCL unlabelled dataset. Despite the fact that the size of the two datasets is comparable, we find that while creating clusters from an irrelevant domain does gain on the baseline (+0.25%), the improvement for clusters built from the relevant domain texts is noticeably higher (+0.6%). The difference between the accuracy on the legal texts and the accuracy on the reviews and weblogs texts is significant at the $p < 0.05$ level.

	CoNLL09	**OntoNotes**
Legal texts	90.19	77.77
Reviews and Weblogs	**90.30**	**78.14***

Table 8: The effect of the domain of unlabelled data on word clusters, discarding word types with count less than 3.

4.3 External lexicon

It is possible to supply to the dependency parser an external lexicon, where word forms are provided with PoS tags. Wiktionary, a companion project for Wikipedia that aims to produce a free, large-scale multilingual dictionary, is a large and constantly growing crowd-sourced resource that appears attractive for NLP research. Wiktionary encodes word definitions, pronunciation, translations, etymology, word forms and part-of-speech information. PoS tag dictionaries derived from Wiktionary have been previously used for out-of-domain PoS tagging (Li et al., 2012) and for PoS tagging of resource-poor languages (Täckström et al., 2013).

To create a lexicon for the parser, we extracted 753,970 English word forms and their PoS tags from a dump of Wiktionary[4]. Wiktionary uses a rather informal set of PoS labels; to convert them to the CoNLL09 tag set, we manually aligned all unique PoS tags found in Wiktionary with those of the CoNLL09 tag set. We compared the accuracy achieved by the parser when the lexicon was supplied, as well as when the lexicon was supplied together with the best configuration word clusters (800 clusters built from the entire SANCL dataset after filtering words with the count less than 3). Table 9 shows results achieved with these settings in comparison to the baseline (improvements on the baseline are indicated with stars). When the lexicon is used on its own, we observe only slight gains on the baseline, on both in-domain and out-domain data, and neither are statistically significant. When combining the lexicon and word clusters, the accuracy actually decreases compared to using word clusters on their own.

Thus the best combination of domain adaptation techniques so far included the use of 800 word clusters built from the entire SANCL unlabelled dataset, after filtering out word forms with the count less than 3, with text normalization, but without the Wiktionary lexicon (+1.15% on the baseline).

[3] http://www.epo.org/searching/free/espacenet.html

[4] http://wiki.dbpedia.org/Wiktionary

	CoNLL09	**OntoNotes**
Wiktionary	90.22	77.73
Clusters	**90.62****	**78.69****
Wiktionary+Clusters	90.44	78.49**
Baseline	90.07	77.54

Table 9: The effect of the Wiktionary lexicon on parsing accuracy.

4.4 Co-Training

Following Sagae (2007), the overall approach to parser co-training we adopted was as follows. First, several parsers were combined to generate additional training data from unlabelled data, i.e. were used as **source learners** for co-training. Then, the Mate parser was re-trained on the augmented training set and tested on a test set, i.e. used as the **evaluation learner**. The reason Mate was selected the evaluation learner was that it achieved the best results on the test data in its default settings (see Table 10).

	CoNLL09	**OntoNotes**
Mate	90.07	77.54
MST	86.9	75.35
Turbo	85.94	74.85
Malt	84.72	72.63

Table 10: The baselines of parsers used in co-training experiments.

Agreement-based co-training. We first experimented with three pairwise parser combinations: using Mate as one source learner and each of the other three parsers as the other source learner in order to obtain additional training data. If two learners agreed on the parse of an unlabelled sentence, i.e. assigned each word form the same dependency label and attached it to the same head, this was taken as an indication of a correct parse, and the sentence was added to the training set. We experimented with different amounts of the additional training sentences added to the main training set in such a manner: 10k, 20k, and 30k sentences. The results of these experiments are shown in Table 11 (significant differences to the baseline results are indicated by stars). The best result is obtained by Mate Malt pair, which outperforms the baseline by just above 1%.

	+10k	**+20k**	**+30k**
Mate+Malt	**78.22****	**78.61****	**78.61****
Mate+MST	78.10**	78.23**	78.31**
Mate+Turbo	77.94**	77.84*	77.99**
Baseline	77.54		

Table 11: Agreement-based co-training using two parsers.

Removing short sentences from unlabelled data. We noticed that among those sentences where two parsers agreed, many tended to be very short: the average number of tokens in generated additional training data was 8 per sentence, while both the training and test set contain much longer sentences on average: the OntoNotes test set had 19.6 tokens/sentence and the CoNLL09 training set had 24.4 tokens/sentence. Such short sentences in the additional training data may be less useful or even harmful for learning an accurate model of the target domain, than those that approximate both training and test data. We experimented with several thresholds (4, 5, and 6 tokens) on the sentence length below which sentences were removed from the additional training data. Table 12 shows that discarding short sentences did improve accuracy by up to 0.25%, though none of the improvements were significant.

Three learners co-training. In the previous experiments, the Mate parser was used both as a **source learner** and as the **evaluation learner**. Therefore it was likely that the additional training data did not

	Mate+Malt, +30k	**Avg. Length**
>6 tokens	**78.88**	13.1
>5 tokens	78.61	12.67
>4 tokens	78.67	11.94
All sentences	78.61	8.35

Table 12: The effect of removing short sentences from generated training data.

contain sufficiently novel examples based on which the evaluation parser could adapt better to the new domain. Thus we next tried the tri-training algorithm (Zhou and Li, 2005), where two parsers are used as source learners and a third as the evalaution learner. We used Malt and MST as source learners, identifying sentences which they parsed in the same manner, and using these sentences to retrain the Mate parser. We find that the tri-training algorithm performs better than the set-up with two parsers: on 10k and 20k additional sentences, it achieves an accuracy increase on Mate+Malt, significant at the $p < 0.05$ level (see Table 13).

	+10k	**+20k**	**+30k**
Mate+Malt+MST	78.70*	**79.12***	78.95
Mate+Malt	78.43	78.70	78.88

Table 13: Accuracy scores for tri-training (Mate+Malt+MST) and the best two-parser co-training algorithm (Mate+Malt).

5 Combining co-training with clusters and an external lexicon

5.1 OntoNotes test set

We explored several possibilities to combine co-training with word clusters and an external lexicon, each time supplying word clusters and/or the lexicon to the Mate parser when it is being retrained on additional training data and applied to the test data. The following configurations of each of the techniques were used:

- Word clusters: 800 clusters generated from the entire SANCL unlabelled dataset, after discarding word types with the count less than 3.
- Lexicon: Wiktionary
- Co-training: Retraining the Mate parser on the combination of initial training set and 20k automatically parsed sentences (agreed by Malt and MST) which contained more than 6 tokens.

The results showed that all three combinations failed to obtain significant improvements over co-training alone. The best result is achieved by combining co-training and clusters, which obtains an increase of only 0.09% on co-training; this is however, the greatest overall improvement on the baseline (+1.67%). The combination of co-training and a Wiktionary lexicon in fact harms accuracy (see Table 14).

5.2 SANCL test set

In order to compare different technique combinations with the results achieved by participants of the SANCL-2012 shared task, we evaluated them on the SANCL test set[5].

As the results in Table 15 indicate, similarly to the results on OntoNotes, word clusters usually fare much better than the Wiktionary-based lexicon, while the latter fails to produce statistically significant

[5] Note that the data was annotated in the Stanford format.

	OntoNotes
Co-training	79.12**
Clusters	78.69**
Wiktionary	77.73
Co-training+Clusters	**79.21****
Co-training+Wiktionary	78.89*
Co-training+Clusters+Wiktionary	79.19**
Baseline	77.54

Table 14: Combination of co-training with word clusters and an external lexicon, OntoNotes test set.

improvements on the baseline. The best accuracy overall was achieved by combinations of techniques, in all the three subdomains, improving on the baseline by up to 1.3%.

Comparing the results achieved by our best configurations with the results of the shared task, we see that our labelled accuracy averaged across the subdomains was just above the Stanford-2 system (80.31 vs. 80.25), which ranked 5th of all the twelve submissions (Petrov and McDonald, 2012). Although our results are still 3.15% lower than DCU-Paris13, the best system at SANCL-2012, the top four results were all generated by combination systems (Le Roux et al., 2012; Zhang et al., 2012; McClosky et al., 2012); our highest results only produced by the Mate parser, hence our best configuration achieved the best performance of a single parser.

	Answers	**Newsgroups**	**Reviews**	**Average**
Co-training	77.18	82.72**	78.21	79.37
Clusters	78.04**	83.06*	79.03**	80.04
Wiktionary	77.61	82.8	78.32	79.57
Clusters+Wiktionary	78.19**	**83.38***	**79.36****	**80.31**
Co-training+Clusters	78.05*	83.29**	78.8**	80.04
Co-training+Clusters+Wiktionary	**78.33****	83.35**	78.84*	80.17
Baseline	77.03	82.4	78.12	79.18
SANCL Stanford-2	77.5	83.56	79.7	80.25
SANCL Best (DCU-Paris13)	81.15	85.38	83.86	83.46

Table 15: Combination of co-training with word clusters and an external lexicon, SANCL test set.

The results on both the OntoNotes and SANCL datasets show that on their own, word clusters and co-training often improve significantly on the baseline, but their combination results only in minor further improvements (only up to 0.32%). Word clusters aim specifically to deal with the unknown vocabulary problem, and, since there seem to be no major grammatical differences between the train and test domains (see Section 3.3), it is likely that the main benefit derived from co-training is the compensation for unknown domain vocabulary. Word clusters also seem a better way to approach this problem: they perform better than co-training on three out of four subdomains. The explanation that unknown vocabulary is the main issue for domain adaptation in this domain pair is further supported by the fact that combinations of word clusters with a Wiktionary lexicon sometimes performed better than combinations involving co-training (on `newsgroups` and `reviews`).

6 Conclusion

In this paper we described experiments with several domain adaptation techniques, in order to quickly adapt a general-language parser to parse web data. We find that the best combination of the techniques improves significantly on the baseline (up to 1.67%), and achieves very promising results on the SANCL-2012 shared task data, outperforming all submissions that used a single parser, in terms of labelled accuracy score averaged across three test sets.

Our experiments with word clusters showed that word clusters derived from unlabelled domain texts consistently contribute to a greater parsing accuracy, and that both the domain relevance of the unlabelled data and its quantity are major factors for successful exploitation of word clusters. Experiments with a crowd-sourced PoS lexicon however were not as conclusive: whereas supplying the lexicon to the parser often resulted in certain accuracy gains, they were not as large as those for word clusters. This suggests word clusters created automatically from relevant domain texts are a better tool to deal with unknown vocabulary than a generic hand-crafted and wide-coverage lexicon. Another interesting finding was that co-training was most effective when the evaluation parser was not used for creating extra training data (the so-called tri-training technique), and when removing very short sentences from automatically labelled data before re-training the evaluation parser.

With respect to combining co-training with word clusters, we could not find clear evidence for additive improvement. This suggests that co-training solves largely the same problem as word clusters, i.e., unknown target domain vocabulary, and that for the web texts under study unknown vocabulary is a much more significant impediment for domain adaptation than grammatical differences between domains.

Acknowledgements

The research was supported by FP7 ICT project "Workbench for Interactive Contrastive Analysis of Patent Documentation" under grant no. FP7-SME-606163.

References

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas Filip Ginter, and Jan Hajic. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Associtation for Computational Linguistics*, 1.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Alicia Burga, Joan Codina, Gabriella Ferraro, Horacio Saggion, and Leo Wanner. 2013. The challenge of syntactic dependency parsing adaptation for the patent domain. In *ESSLLI-13 Workshop on Extrinsic Parse Improvement*.

Marie Candito, Enrique Henestroza Anguiano, and Djam Seddah. 2011. A word clustering approach to domain adaptation: Effective parsing of biomedical texts. In *IWPT*, pages 37–42. The Association for Computational Linguistics.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *HLT-NAACL*, pages 381–384. The Association for Computational Linguistics.

Phani Gadde, L. V. Subramaniam, and Tanveer A. Faruquie. 2011. Adapting a WSJ trained part-of-speech tagger to noisy text: Preliminary results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, MOCRAND11, pages 51–58, New York, NY, USA. ACM.

Daniel Gildea. 2001. Corpus variation and parser performance. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, EMNLP '01, pages 167–202, Stroudsburg. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *16th Nordic Conference of Computational Linguistics*, pages 105–112. University of Tartu.

Mohammad Khan, Markus Dickinson, and Sandra Kübler. 2013a. Does size matter? text and grammar revision for parsing social media data. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 1–10, Atlanta, Georgia, June. Association for Computational Linguistics.

Mohammad Khan, Markus Dickinson, and Sandra Kübler. 2013b. Towards domain adaptation for parsing web data. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov, editors, *RANLP*, pages 357–364. RANLP 2011 Organising Committee / ACL.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *In Proc. ACL/HLT*.

Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. Dcu-paris13 systems for the sancl 2012 shared task.

Matthew Lease and Eugene Charniak. 2005. Parsing biomedical literature. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *IJCNLP*, volume 3651 of *Lecture Notes in Computer Science*, pages 58–69. Springer.

Shen Li, Joo Graa, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *EMNLP-CoNLL*, pages 1389–1398. ACL.

Percy Liang. 2005. Semi-supervised learning for natural language. In *MASTERS THESIS, MIT*.

André FT Martins, Noah A Smith, Eric P Xing, Pedro MQ Aguiar, and Mário AT Figueiredo. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44. Association for Computational Linguistics.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *ACL (Short Papers)*, pages 101–104. The Association for Computer Linguistics.

David McClosky, Wanxiang Che, Marta Recasens, Mengqiu Wang, Richard Socher, and Christopher Manning. 2012. Stanfords system for parsing the english web. In *Workshop on the Syntactic Analysis of Non-Canonical Language (SANCL 2012). Montreal, Canada*.

Ryan T McDonald and Fernando CN Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL*.

Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 915–932. sn.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 351–359. Association for Computational Linguistics.

Lilja Øvrelid and Arne Skjærholt. 2012. Lexical categories for improved parsing of web data. In *Proceedings of COLING 2012: Posters*, pages 903–912, Mumbai, India, December. The COLING 2012 Organizing Committee.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*, pages 380–390. The Association for Computational Linguistics.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, volume 59.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 705–713, Stroudsburg, PA, USA. Association for Computational Linguistics.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1566–1576. Association for Computational Linguistics.

Sampo Pyysalo, Tapio Salakoski, Sophie Aubin, and Adeline Nazarenko. 2006. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3).

Ines Rehbein. 2011. Data point selection for self-training. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, SPMRL '11, pages 62–67, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenji Sagae. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *In Proceedings of the Eleventh Conference on Computational Natural Language Learning*.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anders Søgaard and Barbara Plank. 2012. Parsing the web as covariate shift. In *Workshop on the Syntactic Analysis of Non-Canonical Language (SANCL2012)*, Montreal, Canada.

Mark Steedman, Anoop Sarkar, Miles Osborne, Rebecca Hwa, Stephen Clark, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *EACL*, pages 331–338. The Association for Computer Linguistics.

Peter Szolovits. 2003. Adding a medical lexicon to an English parser. In *AMIA Annual Symposium Proceedings*, volume 2003, page 639. American Medical Informatics Association.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan T. McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *TACL*, 1:1–12.

Meishan Zhang, Wanxiang Che, Yijia Liu, Zhenghua Li, and Ting Liu. 2012. Hit dependency parsing: Bootstrap aggregating heterogeneous parsers. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *Knowledge and Data Engineering, IEEE Transactions on*, 17(11):1529–1541.

Guangyou Zhou, Jun Zhao, Kang Liu, and Li Cai. 2011. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1556–1565, Stroudsburg, PA, USA. Association for Computational Linguistics.