Towards Domain Adaptation for Parsing Web Data

Mohammad Khan
Indiana University
Bloomington, IN USA
khanms@indiana.edu

Markus Dickinson
Indiana University
Bloomington, IN USA
md7@indiana.edu

Sandra Kübler
Indiana University
Bloomington, IN USA
skuebler@indiana.edu

Abstract

We improve upon a previous line of work for parsing web data, by exploring the impact of different decisions regarding the training data. First, we compare training on automatically POS-tagged data vs. gold POS data. Secondly, we compare the effect of training and testing within sub-genres, i.e., whether a close match of the genre is more important than training set size. Finally, we examine different ways to select out-of-domain parsed data to add to training, attempting to match the in-domain data in different shallow ways (sentence length, perplexity). In general, we find that approximating the in-domain data has a positive impact on parsing.

1 Introduction and Motivation

Parsing data from the web is notoriously difficult, as parsers are generally trained on news data (Petrov and McDonald, 2012). The problem, however, varies greatly depending upon the particular piece of web data: what is often termed web data is generally a combination of different sub-genres, such as Facebook posts, Twitter feeds, YouTube comments, discussion forums, blogs, etc. The language used in such data does not follow standard conventions in various respects (see Herring, 2011): 1) The data is edited to varying degrees, with Twitter on the lower end and professional emails and blog on the upper end of the scale. 2) The sub-genres often display characteristics of spoken language, including sentence fragments and colloquialisms. 3) Some web data, especially social media data, typically contains a high number of emoticons and acronyms such as *LOL*.

At the same time, there is a clear need to develop basic NLP technology for a variety of types of web data. To perform tasks such as sentiment analysis (Nakagawa et al., 2010) or information extraction (McClosky et al., 2011), it helps to part-of-speech (POS) tag and parse the data, as a step towards providing a shallow semantic analysis.

We continue our work (Khan et al., 2013) on dependency parsing web data from the English Web Treebank (Bies et al., 2012). We previously showed that text normalization has a beneficial effect on the quality of a parser on web data, that we can further improve the parser's accuracy by a simple, n-gram-based parse revision method, and that having a balanced training set of out-of-domain and in-domain data provides the best results when parsing web data. The current work extends this previous work by more closely examining the data given as input for training the parser. Specifically, we take the following directions:

1. All previous experiments were carried out on gold part of speech (POS) tags. Here, we investigate using a POS tagger trained on out-of-domain data, thus providing a more realistic setting for parsing web data. We specifically test the impact of training the parser on automatic POS tags (section 4).

2. The web data provided in the English Web Treebank (EWT) is divided into five different sub-genres: 1) answers to questions, 2) emails, 3) newsgroups, 4) reviews, and 5) weblogs. Figure 1 shows examples from the different sub-genres. So far, we used the whole set across these genres, which raises questions about whether a closer match of the genre is more important than the data size, and we thus investigate parsing results within

Proceedings of Recent Advances in Natural Language Processing, pages 357–364,
Hissar, Bulgaria, 7-13 September 2013.

1. **Answer:** where can I get morcillas in tampa bay , I will like the argentinian type , but I will try anothers please ?
2. **Email:** Michael : <s> Thanks for putting the paperwork together . <s> I would have interest in meeting if you can present unique investment opportunities that I do n't have access to now .
3. **News:** complete with original Magnavox tubes - all tubes have been tested they are all good - stereo amp
4. **Review:** Buyer Beware !! <s> Rusted out and unsafe cars sold here !
5. **Blog:** The Supreme Court announced its ruling today in Hamdan v. Rumsfeld divided along ideological lines with John Roberts abstaining due to his involvement at the D.C. Circuit level and Anthony Kennedy joining the liberals in a 5 - 3 decision that is 185 pages long .

Figure 1: Example sentences from each sub-genre (<s> = sentence boundary)

each sub-genre, and whether adding easy-to-parse data to training improves performance for the difficult sub-genres (section 5).

3. Finally, from our previous work, we know that combining the EWT training set with sentences from the Penn Treebank is beneficial. However, we do not know how to best select the out-of-domain sentences. Should they be drawn randomly; should they match in size; should the sentences match in terms of parsing difficulty (cf. perplexity)? We explore different ways to match the in-domain data (section 6).

2 Related Work

There is a growing body of work on parsing web data, as evidenced by the 2012 Shared Task on Parsing the Web (Petrov and McDonald, 2012). There have been many techniques employed for improving parsing models, including normalizing the potentially ill-formed text (Foster, 2010; Gadde et al., 2011; Øvrelid and Skjærholt, 2012) and training parsers on unannotated or reannotated data, e.g., self-training or uptraining, (e.g., Seddah et al., 2012; Roux et al., 2012; Foster et al., 2011b,a). Less work has gone into investigating the impact of different genres or on specific details of the sentences given to the parser.

Indeed, Petrov and McDonald (2012) mention that for the shared task, "[t]he goal was to build a single system that can robustly parse all domains, rather than to build several domain-specific systems." Thus, parsing results were not obtained by genre. However, Roux et al. (2012) demonstrated that using a genre classifier, in order to employ specific sub-grammars, helped improve parsing performance. Indeed, the quality and fit of data has been shown for in-domain parsing (e.g. Hwa, 2001), as well as for other genres, such as questions (Dima and Hinrichs, 2011).

One common, well-documented ailment of web parsers is the effect of erroneous tags on POS accuracy. Foster et al. (2011a,b), e.g., note that propagation of POS errors is a serious problem, especially for Twitter data. Researchers have thus worked on improving POS tagging for web data, whether by tagger voting (Zhang et al., 2012) or word clustering (Owoputi et al., 2012; Seddah et al., 2012). There are no reports about the impact of the quality of POS tags for training—i.e., whether worse, automatically-derived tags might be an improvement over gold tags—though Søgaard and Plank (2012) note that training with predicted POS tags improves performance.

Researchers have trained parsers using additional data which generally fits the testing domain, as mentioned above. There has been less work, however, on extracting specific types of sentences which fit the domain well. Bohnet et al. (2012) noticed a problem with parsing fragments and so extracted longer NPs to include in training as standalone sentences. From a different perspective, Søgaard and Plank (2012) weight sentences in the training data rather than selecting a subset, to better match the distribution of the target domain. In general, identifying sentences which are similar to a particular domain is a concept familiar in active learning (e.g., Mirroshandel and Nasr, 2011; Sassano and Kurohashi, 2010), where dissimilar sentences are selected for hand-annotation to improve parsing.

3 Experimental Setup

3.1 Data

For our experiments, we use two main resources, the Wall Street Journal (WSJ) portion of the Penn Treebank (PTB) (Marcus et al., 1993) and the English Web Treebank (EWT) (Bies et al., 2012). The EWT is comprised of approx. 16 000 sentences from weblogs, newsgroups, emails, reviews, and question-answers. Note that our data sets are different from the ones in Khan et al. (2013) since in the previous work we had removed sentences with POS labels AFX and GW.

To create training and test sets, we broke the data into the following sets:

- WSJ training: sections 02-22 (42 009 sent.)
- WSJ testing: section 23 (2 416 sent.)
- EWT training: 80% of the data, taking the first four out of every five sentences (13 298 sent.)
- EWT testing: 20% of the data, taking every fifth sentence (3 324 sent.)
- EWT sub-genre training and test data: here, we create individual training and test sets for the 5 genres: EWT$_{blog}$, EWT$_{news}$, EWT$_{email}$, EWT$_{review}$, and EWT$_{answer}$, using the same sampling described above

The two corpora were converted from PTB constituency trees into dependency trees using the Stanford dependency converter (de Marneffe and Manning, 2008).[1] Since the EWT uses data that shows many of the characteristics of non-standard language, we decided to normalize the spelling of the EWT training and the test set.

For the normalization, we reduce all web URLs to a single token, i.e., each web URL is replaced with the place-holder URL. Similarly, all emoticons are replaced by a single marker EMO. Repeated use of punctuation, e.g., *!!!*, is reduced to a single punctuation token.

3.2 POS Tagger

We use TnT (Brants, 2000), a Markov model POS tagger using a trigram model. It it is fast to train and has a state-of-the-art model for unknown words, using a suffix trie of hapax legomena.

3.3 Parser

We use MSTParser (McDonald and Pereira, 2006),[2] a freely-available parser that reaches state-of-the-art accuracy in dependency parsing for English. MST is a graph-based parser which optimizes its parse tree globally (McDonald et al., 2005), using a variety of feature sets, i.e., edge, sibling, context, and non-local features, employing information from words and POS tags. We use its default settings for all experiments.

Train	Test	POS acc.
WSJ	WSJ	96.73%
EWT	EWT	**94.28%**
WSJ	EWT	88.73%
WSJ+EWT (balanced)	EWT	93.48%

Table 1: Results of using TnT in and out of domain

3.4 Evaluation

For parser evaluation, we report unlabeled attachment scores (UAS) and labeled attachment scores (LAS), the percentage of dependencies which are attached correctly or attached and labeled correctly (Kübler et al., 2009). Parser evaluation is carried out with MSTParser's evaluation module. For POS tagger evaluation, we report accuracy based on TnT's evaluation script. Significance testing was performed using the CoNLL 2007 shared task evaluation using Dan Bikel's Randomized Parsing Evaluation Comparator.[3]

4 The effect of POS tagging

We here explore the effect of POS tagging on parsing web data, to see how closely the conditions for training should match the conditions for testing.

However, first we need to gauge the effect of using the TnT POS tagger out of domain. For this reason, we conducted a set of experiments, training and testing TnT in different conditions. The results are shown in table 1. They show that TnT reaches an accuracy of 96.7% when trained and tested on the WSJ. This corroborates findings by Brants (2000). When we train TnT on EWT training data, running it on the EWT testing data delivers an accuracy of 94.28%, already 2–3% below performance on news data. However, note that the EWT is much smaller than the full WSJ. In contrast, if we train TnT on WSJ and then use it for POS tagging EWT data, we only reach an accuracy of 88.73%. Even if we balance the source and target domain data, which proved beneficial in our previous experiments on parsing (Khan et al., 2013), we reach an accuracy of 93.48%, well below the in-domain tagging result for the EWT. This means that in contrast to parsing, the POS tagger requires less training data and profits more

[1] http://nlp.stanford.edu/software/stanford-dependencies.shtml

[2] http://sourceforge.net/projects/mstparser/

[3] http://nextens.uvt.nl/depparse-wiki/SoftwarePage

Train	Test	POS acc.	UAS	LAS
Gold	Gold	100%	85.78%	83.14%
Gold	TnT	94.28%	81.89%	77.69%
TnT	TnT	94.28%	***82.52%**	***78.54%**

Table 2: The effect of POS tagging on parser performance, using the base EWT data split (*=sig. at the 0.01 level, as compared to Train=Gold/ Test=TnT)

from the small target domain training set than from a larger training set with out-of-domain data.

Given this degree of error in tagging, a parser trained with similar noise in POS tags may outperform one which is trained on gold tags. Thus, we run TnT on the training data, using a 10-fold split of the training set: each tenth of the training corpus is tagged using a POS tagger trained on the other 9 folds. Then we use the combination of all the automatically POS tagged folds and insert those POS tags into the gold standard dependency trees before we train the parser.

The three conditions for POS tagging are shown in table 2. The first point to note is the impact of switching from gold to automatic POS tags: testing on TnT tags results in a degradation of about 4.5–5.5% in LAS, as compared to gold standard POS tags in the test set, consistent with typical drops in performance (e.g., Rehbein et al., 2012).

More to the point for our purposes, we see in table 2 that training a parser on automatically-assigned POS tags outperforms a parser trained on gold POS tags. LAS increases from 77.69% to 78.54%. This supports the notion that training data should match testing closely. However, it also shows that we need to investigate methods for improving POS tagger accuracy.

5 The effect of domain

As mentioned, the EWT contains subcorpora from five different genres, and, while they share many common features (misspellings, unknown words), they have many unique properties, as illustrated in the examples in figure 1. In terms of sentence length, domains such as weblogs lend themselves more easily to longer, more well-edited sentences, matching news data better. Reviews, on the other hand, often have shorter sentences—similar to, e.g., email greetings. Run-ons are common across genres, but we see them here in the answer and news sub-genres. The example for the answer sub-corpus shows some of the difficult challenges faced by a parser, as it contains a declarative sentence embedded within the question, where the final word (*please*) attaches back to the question.

To gauge the effect of different sub-genres, we trained and tested the parser within each sub-genre. In order to concentrate on the differences in parsing, we used gold POS tags for these experiments. Results for the five individual sub-corpora are given in the first five rows of table 3. It is noteworthy that there is nearly a 5% difference in LAS between the best sub-genre (EWT_{email}) and the worst (EWT_{answer}). We also show various properties of the sub-corpora, including number of tokens (*Tokens*), the average sentence length (*SenLen*), and the number of finite verbal roots (*FinRoot*)[4] in training; and also the percentage of unknown word tokens in the test corpus, as compared to the training corpus (*Unk.*)

In general, emails and reviews fare the best, likely due to a combination of shorter sentences (11.84 and 14.58, respectively) and text that tends to follow grammatical conventions. Blogs and newsgroups are in the middle, with longer, harder-to-parse sentences (18.17 and 22.07, respectively) and higher levels of unknown words in testing (12.2% and 10.2%), but being consistently fairly well-edited. While it might be surprising that the results for these two sub-genres are lower than emails and reviews, note that the training for both domains is significantly lower, on the order of 10,000 words less than the other corpora. It is possible that with more data, these well-edited domains would see improved parser performance.

On the lower end of the parsing spectrum is the domain of answers, which is a curious trend. There is nearly as much training data as with emails and reviews, and the average sentence length is comparable. If we look at the number of non-finite sentence roots—as a way to approximate the number of non-fragment sentences—it is nearly identical to the email sub-genre. We suspect that the fragments are not as systematic as greetings and that users may post replies quickly, leading to less well-formed text, but this deserves future consideration.

Given the poor performance on the answer domain and the higher performance of the parser on

[4] The Stanford converter treats the predicate as the head of copular sentences, e.g., a noun or adjective; thus, the number of finite roots does not correspond directly to the number of non-fragmentary sentences.

Train	Tokens	Sen-Len	Fin-Root	Test	Unk.	UAS	LAS
EWT$_{answer}$	43 173	15.47	767	EWT$_{answer}$	8.2%	81.25%	78.03%
EWT$_{email}$	46 473	11.85	765	EWT$_{email}$	8.0%	85.04%	82.82%
EWT$_{news}$	34 762	18.17	558	EWT$_{news}$	12.2%	81.65%	79.12%
EWT$_{review}$	44 483	14.58	1 048	EWT$_{review}$	8.5%	82.92%	79.64%
EWT$_{blog}$	35 868	22.07	635	EWT$_{blog}$	10.2%	81.68%	79.00%
EWT$_{answer}$+EWT$_{email}$	89 646	13.36	1 532	EWT$_{answer}$	6.5%	**82.16%	**79.05%
EWT$_{answer}$+EWT$_{news}$	77 935	16.57	4571	EWT$_{answer}$	6.3%	**82.84%	**79.59%
EWT$_{answer}$+EWT$_{blog}$	79 041	17.90	4874	EWT$_{answer}$	6.5%	**82.53%	**79.43%
EWT$_{answer}$+EWT$_{balanced}$	102 717	19.13	1 482	EWT$_{answer}$	5.7%	**83.07%	**79.74%
EWT$_{answer}$+EWT$_{rest}$	204 759	19.24	12 312	EWT$_{answer}$	4.4%	****84.01%**	****80.97%**

Table 3: The effect of domain on parser performance, using gold POS tags (** = sig. at the 0.01 level, testing all conditions below the line, as compared to the first row Train=EWT$_{answer}$)

emails, we decided to see whether parsing could be improved by adding data to the small answer training set 1) from the domain that is easiest to parse: emails, 2) from the news domain because of its similar average sentence length, and 3) from the blog domain because it has the longest sentences. We compare these configurations with one where we add the same number of sentences, but sampled from all four remaining domains (*balanced*) and one where we add all the training data from all other genres (*rest*). We see a clear improvement for all settings, in comparison with using only the answer data for training. The best results are obtained by using all other genres as additional training data, showing that the size of the training set is the most important variable.

The results also show that the sampling from all remaining sub-genres results in higher parsing accuracy than just using the easiest to parse data set, illustrating that we should not look for data which is generally easy to parse, but data which is the best fit for the test data.

6 The effect of sentence selection

In our previous work (Khan et al., 2013), we showed that we obtain the best results when we use a balanced training corpus with the same number of sentences from the EWT and the WSJ. On the one hand, these results show that in-domain data is critical for the success of the parser; on the other hand, out-of-domain data is important to increase the size of the training set. It is thus important to find a good balance between using more training data and not overpowering the in-domain data. This leads to the question of whether it is possible to choose sentences from an out-of-domain data set that are similar to the sentences in the target domain rather than just selecting a portion of consecutive sentences. In other words, can we identify sentences from the WSJ that will have the best impact on a parser for web data?

Train	Tokens	UAS	LAS
EWT+WSJ	1 205 621	85.73%	83.12%
EWT+WSJSent	524 236	**86.34%**	**83.83%**
EWT+WSJToken	399 915	86.26%	83.69%
EWT+WSJDist	424 297	**86.34%**	83.73%
EWT+LowP	619 591	***86.68%**	****84.20%**
EWT+AllLowP	819 856	*86.64%	*84.08%
EWT+MedLowP	568 666	86.41%	83.85%
EWT+MidP	529 936	86.13%	*83.54%

Table 4: The effect of selection on parser performance: all experiments on EWT testing data with gold POS tags; WSJ data defined in the text (*/** = sig. at the 0.05/0.01 level, testing the 4 perplexity models as compared to EWT+WSJSent)

In the first set of experiments, we investigate simple heuristics to choose a good set of training sentences from the WSJ: In the first experiment, we use the full WSJ (*EWT+WSJ*). Then we restrict the WSJ part to match the number of sentences from the EWT (*EWT+WSJSent*). However, since WSJ sentences are longer on average than EWT sentences, we repeat the experiment but choose the WSJ subset so that it matches the number of words in the EWT training set (*EWT+WSJToken*). Finally, we choose the WSJ sentences so that they match the distribution of sentence lengths in EWT (*EWT+WSJDist*). For example, if EWT has 100 sentences with 10 words, we select 100 sentences of length 10 from the WSJ. All of these experiments are again carried out with gold POS tags.

The results of these experiments are shown in the first two parts of table 4. The results for the selection methods show that selecting the WSJ part based on the number of words results in the lowest parsing accuracy. Choosing the WSJ part based on the number of sentences or the distribution of sentence length results in the same unlabeled accuracy (UAS) of 86.34%, as compared to 86.26% for the word based selection. However, the selection based on the number of sentences results in a higher labeled accuracy of 83.83%, as opposed to 83.73% for the distribution of sentence length. We suspect that the random selection of sentences gives more variety, which is beneficial for training. However, note that the difference in the number of words in the training set across these three methods is minimal: they vary only by 41 words.

In a second set of experiments, we decided to use a more informed method for choosing similar sentences: perplexity. Thus, we trained a language model on the (stemmed) words of the test set based on a 5-gram word model, and then calculated perplexity for each sentence in the WSJ, normalized by the length of the sentence. We used the *CMU-Cambridge Statistical Language Modeling Toolkit*[5] for calculating perplexity. Perplexity should give an approximation of distance between sentences in the two corpora. We experimented with different selection strategies:

1. Low Perplexity (*LowP*): We select the sentences with the lowest perplexity, i.e., the most similar ones to the test set; we restricted the number of sentences from the WSJ to match the size of the EWT training set.
2. All Low Perplexity (*AllLowP*): Here, we also selected sentences with low perplexity, but this time used all sentences below the median, i.e. half the WSJ sentences.
3. Low Perplexity close to the median (*MedLowP*): Here, we investigate the effect of choosing sentences that are less similar to the test sentences: we select the same number of sentences as with LowP, but this time from the median down. In other words, the sentences with the lowest perplexity, i.e., the most similar sentences, are excluded. This is based on the assumption that if the chosen sentences are too similar, it will not have much effect on the trained model.
4. Mid-range Perplexity (*MidP*): In this set, we choose sentences that are even less similar to the test sentences. We again choose the same number of sentences as in the EWT training set, but half of them from the median and down and half from the median up.

The results are in the final four rows of table 4. Interestingly, the best-performing method adds low-perplexity data to training. Thus, selecting data which is more similar to the domain helps the most. Furthermore, once the data is farther away, it starts to harm parsing performance, as can be seen in the (albeit minimal) difference between the EWT+LowP and EWT+AllLowP models.

7 Summary and Outlook

Exploring the parsing of web data, we have investigated different decisions that go into the training data, demonstrating how the better the fit of the training data to the testing data—in properties ranging from the nature of the POS tags to which sentences go into the data—the better performance the parser will have. We first compared training on automatically POS-tagged data vs. gold POS tag data, showing that performance improves by automatically tagging the training data. Next, we compared the effect of training and testing within sub-genres and saw that features such as sentence length have a strong effect. Finally, we examined ways to select out-of-domain parsed data to add to training, attempting to match the in-domain data in different shallow ways, and we found that matching training sentences to a language model improves parsing. In short, fitting the training data to the in-domain data, in even fairly superficial ways, has a positive impact on parsing results.

There are several directions to take this work. First, the sentence selection methods, for example, can be combined with self-training techniques to not only increase the training data size, but to only add sentences which fit the test domain well. Secondly, the work on understanding sub-genres of web parsing deserves more thorough treatment in the future to tease apart which components are most problematic (e.g., sentence fragments), how they can be automatically identified, and how the parser can be adjusted to accommodate them.

[5] http://www.speech.cs.cmu.edu/SLM_info.html

References

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. Linguistic Data Consortium, Philadelphia, PA.

Bernd Bohnet, Richard Farkas, and Ozlem Cetinoglu. 2012. SANCL 2012 shared task: The IMS system description. In *Workshop on the Syntactic Analysis of Non-Canonical Language (SANCL 2012)*. Montreal, Canada.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP)*, pages 224–231. Seattle, WA.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*. Manchester, England.

Corina Dima and Erhard Hinrichs. 2011. A semi-automatic, iterative method for creating a domain-specific treebank. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 413–419. Hissar, Bulgaria.

Jennifer Foster. 2010. "cba to check the spelling": Investigating parser performance on discussion forum posts. In *Proceedings of NAACL-HLT 2010*, pages 381–384. Los Angeles, CA.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011a. #hardtoparse: POS tagging and parsing the twitterverse. In *The AAAI-11 Workshop on Analyzing Microtext*. San Francisco.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011b. From news to comment: Resources and benchmarks for parsing the language of Web 2.0. In *Proceedings of IJCNLP-11*, pages 893–901. Chiang Mai, Thailand.

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 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*. Beijing, China.

Susan Herring. 2011. Discourse in Web 2.0: Familiar, reconfigured, and emergent. In *Georgetown University Round Table on Languages and Linguistics 2011: Discourse 2.0: Language and New Media*. Washington, DC.

Rebecca Hwa. 2001. On minimizing training corpus for parser acquisition. In *Proceedings of the Workshop on Computational Natural Language Learning (CoNLL)*. Toulouse, France.

Mohammad Khan, Markus Dickinson, and Sandra Kübler. 2013. Does size matter? Text and grammar revision for parsing social media data. In *Proceedings of the NAACL Workshop on Language Analysis in Social Media*. Atlanta, GA.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Morgan & Claypool Publishers.

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, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL-HLT-11*, pages 1626–1635. Portland, OR.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL-05*, pages 91–98. Ann Arbor, MI.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL-06*. Trento, Italy.

Seyed Abolghasem Mirroshandel and Alexis Nasr. 2011. Active learning for dependency parsing using partially annotated sentences. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 140–149. Dublin, Ireland.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of NAACL-HLT 2010*, pages 786–794. Los Angeles, CA.

Lilja Øvrelid and Arne Skjærholt. 2012. Lexical categories for improved parsing of web data. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 903–912. Mumbai, India.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2012. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL 2013*. Atlanta, GA.

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

Ines Rehbein, Hagen Hirschmann, Anke Lüdeling, and Marc Reznicek. 2012. Better tags give better trees - or do they? *Linguistic Issues in Language Technology (LiLT)*, 7(10).

Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad Zadeh Kaljahi, and Anton Bryl. 2012. DCU-Paris13 systems for the SANCL 2012 shared task. In *Workshop on the Syntactic Analysis of Non-Canonical Language (SANCL 2012)*. Montreal, Canada.

Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for Japanese dependency parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 356–365. Uppsala, Sweden.

Djamé Seddah, Benoit Sagot, and Marie Candito. 2012. The alpage architecture at the sancl 2012 shared task: Robust pre-processing and lexical bridging for user-generated content parsing. In *Workshop on* *the Syntactic Analysis of Non-Canonical Language (SANCL 2012)*. Montreal, Canada.

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

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