Overview of the 2012 Shared Task on Parsing the Web

Slav Petrov and Ryan McDonald
Google
New York, NY
{slav|ryanmcd}@google.com

Abstract

We describe a shared task on parsing web text from the Google Web Treebank. Participants were to build a single parsing system that is robust to domain changes and can handle noisy text that is commonly encountered on the web. There was a constituency and a dependency parsing track and 11 sites submitted a total of 20 systems. System combination approaches achieved the best results, however, falling short of newswire accuracies by a large margin. The best accuracies were in the 80-84% range for F1 and LAS; even part-of-speech accuracies were just above 90%.

1 Introduction

The field of syntactic parsing has seen much progress over the last two decades. As accuracies have improved, parsing promises to become an integral part of downstream applications that can benefit from high accuracy syntactic analysis. When evaluated on standard (newswire) benchmarks, current parsers achieve accuracies well above 90%. However, as has been shown in the past (Petrov et al., 2010; Foster et al., 2011) and is also demonstrated here, these 90%+ accuracies are limited to heavily edited domains. In practice, parsing accuracies are much lower, hovering barely over 80%; even part-of-speech tagging accuracies are often in the low 90ies.

Most applications that rely on parsing, such as machine translation, sentiment analysis and information extraction need to handle, more often than not, unedited text. These texts often come from domains common on the web such as blogs, discussion forums, consumer reviews, etc. In order to reliably translate and extract information from the web, progress must be made in parsing such texts.

There are multiple reasons that parsing the web is difficult, all of which stem from a mismatch with the training data, which is typically the Wall Street Journal (WSJ) portion of the Penn Treebank. Punctuation and capitalization are often inconsistent, making it difficult to rely on features that can be predictive for newswire. There is often a lexical shift due to increased use of slang, technical jargon or other phenomena. Spelling mistakes and ungrammatical sentences are not uncommon. Another important factor is that some syntactic constructions are more frequent in web text than in newswire: most notably questions, imperatives, long lists of names and sentence fragments.

The recently constructed Google Web Treebank (Section 2) provides a manually annotated corpus for such noisy web texts. It covers five domains and provides a large set of unlabeled text from each domain, and smaller sets ($\approx$ 2,000-4,000 sentences per domain) annotated with syntactic parse trees in the style of Ontonotes 4.0. The Google Web Treebank can be used as a large, high quality test set, making the hitherto standard test set (WSJ Section 23) obsolete after 20 years of intensive use. The primary purpose of the large corpus of unlabeled sentences provided with the treebank is to help make semi-supervised learning and domain adaptation tangible.

In this work we describe a shared task on parsing web text that was held during the spring of 2012. As described in more detail in Section 3, participants were provided with the Ontonotes release of the WSJ treebank and the unlabeled portion of the Google Web Treebank for training. Two of the la-

	Training	Development			Evaluation			
	WSJ-train	Emails	Weblogs	WSJ-dev	Answers	Newsgroups	Reviews	WSJ-eval
Sentences	30,060	2,450	1,016	1,336	1,744	1,195	1,906	1,640
Tokens	731,678	29,131	24,025	32,092	28,823	20,651	28,086	35,590
Types	35,933	5,478	4,747	5,889	4,370	4,924	4,797	6,685
OOV	0.0%	30.7%	19.6%	11.8%	27.7%	23.1%	29.5%	11.5%

Table 1: Training, development and evaluation data statistics. Shown are the number sentences, tokens and unique types in the data. OOV is the percentage of tokens in the data set that are not observed in WSJ-train.

	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
Types	221,576	166,515	33,325	357,090	287,575

Table 2: Statistics for unlabeled data without any text normalization. These statistics are approximate as both sentence splitting and tokenization were automatically applied to these data sets.

beled domains were used during development, while the remaining three were reserved for the final evaluation. 11 sites participated in the shared task, submitting 8 constituency and 12 dependency parsing systems (Section 4). Many systems built on top of publicly available parsers, potentially combining multiple models, and making use of the unlabeled data via self-training and word clusters.

Overall, system combination approaches produced the highest accuracies, which however were only in the 80-84% range. This is still a positive result as the baseline parsers – both of which are state-of-the-art constituency and dependency systems – only achieved accuracies in the 75-80% range. We found that there were good correlations between the performance on the various test domains from the web, but the correlation to the WSJ newswire data was much weaker. The results also show that web data poses significant problems to current part-of-speech taggers, which achieve accuracies of just above 90% on the web data. We discuss some additional trends in Section 5.

2 Google Web Treebank

The Google Web Treebank covers five domains: Yahoo! Answers, Emails, Newsgroups, Local Business Reviews and Weblogs. These domains were chosen to cover the different genres commonly found on the web, as well as different syntactic and stylistic variations found in written language.

For each domain, first a large set of data was collected (in most cases more than 1 million sentences). A much smaller subset was then randomly sampled at the document level and manually annotated with syntactic parse trees in the style of Ontonotes 4.0 by professional annotators from the Linguistic Data Consortium (LDC). We additionally subdivided the labeled data for each domain into a development and an evaluation half. Statistics for the data used in the shared task, both labeled and unlabeled, can be seen in Table 1 and Table 2.

Since the Google Web Treebank is natively a constituency treebank, we used version 2.0 of the Stanford converter (De Marneffe et al., 2006) to convert it to labeled dependencies. Figure 1 shows a constituency tree from the email domain and its converted dependency version. This converter has been primarily developed on the Penn Treebank. As a result, it is quite robust on WSJ newswire text. However, there are many cases where the converter breaks down on the web data due to non-conventional or poor writing conventions. The most notable case is for coordination and apposition structures. The Stanford converter typically requires the explicit inclusion of a conjunction to distinguish between these two when a conjunction/apposition can have more than two components. Unfortunately, many lists and conjunctions in the web data do not contain such an explicit conjunction, resulting in many apposition dependency labels that in fact should be conjunctions. Additionally, as pointed out by one of the participants (Pitler, 2012), the structure of the coordination phrase could differ depend-

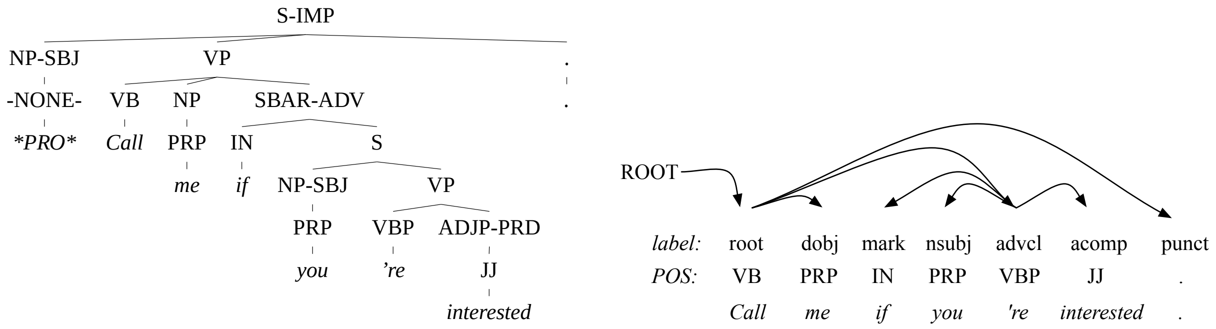
S-IMP
NP-SBJ
VP
.
-NONE-
VB
NP
SBAR-ADV
.
PRO
Call
PRP
IN
S
me
if
NP-SBJ
VP
PRP
VBP
ADJP-PRD
you
're
JJ
interested
ROOT
label: root dobj mark nsubj advcl acomp punct
POS: VB PRP IN PRP VBP JJ .
Call me if you 're interested .

Figure 1: Example constituency and dependency trees. As is common practice in constituency parsing, function labels and empty nodes were available at training time but were not included in the evaluation.

ing on the presence of punctuation at the end of a conjunction, i.e., “X, Y and Z” versus “X, Y, and Z”, or the inconsistent inclusion of NP unary productions. However, instead of systematically or manually correcting such cases we opted to leave them in place for replicability reasons. Thus, if one obtains the Google Web Treebank, then one simply needs to run the data through the version 2.0 of the Stanford converter to get the exact data used in the evaluation.

3 Shared Task

Participants in the shared task were provided with the following sets of data:

1. Sections 02-21 of the WSJ portion of Ontonotes 4.0 (30,060 parsed sentences).
2. Five sets of unlabeled sentences (27,000 to 2,000,000 sentences per domain).
3. Two development sets from the new Google Web Treebank (1,016 parsed sentences from the weblog domain and 2,450 parsed sentences from the emails domain).
4. Section 22 of the WSJ portion of Ontonotes 4.0 (1,336 parsed sentences).

We used the portion of the WSJ from Ontonotes 4.0 and not the full original treebank as Ontonotes 4.0 and the Google Web Treebank share annotation standards.[1] These standards are slightly different from the original PTB in aspects such as tokenization and significantly different in aspects such as noun-phrase bracketing.

The task was to build the best possible parser by using only data sets (1) and (2). Data set (3) was provided as a development set, while the official evaluation set consisted of the remaining three domains of the Google Web Treebank. Data set (4) was provided as an addition reference point for newswire accuracy. The goal was to build a single system that can robustly parse all domains, rather than to build several domain-specific systems. We required all participating systems to only submit results trained on data sets (1) and (2). I.e., we did not allow the addition of other labeled or unlabeled data. In particular the development data sets (3) and (4) were not to be used for training the final system. It was permissible to use previously constructed lexicons, word clusters or other resources provided that they are made available for other participants.

There were two tracks in the shared task, one for constituency parsers and one for dependency parsers. We additionally converted the output of the submitted constituency parsers to dependencies. For the evaluation, the participants were provided with the raw sentences of the test portion of the Yahoo! Answers, Newsgroups and Local Business Reviews domains from the Google Web Treebank. The evaluation data was not annotated with part-of-speech (POS) tags, and the participants were expected to run their own POS tagger either as part of the parser or as a standalone pre-processing component. Additionally, participants were also provided with Section 23 of the WSJ portion of Ontonotes 4.0. The official evaluation was performed only on the web data. We used the WSJ evaluation results to compare in-domain and out-of-domain performance.

[1] http://www.ldc.upenn.edu/Catalog/docs/LDC2011T03/

The submitted system outputs were evaluated using standard tools: evalb for constituent labeled precision (LP), recall (LR) and F1; the CoNLL 2006 eval.pl script for unlabeled (UAS) and labeled attachment score (LAS) (Buchholz and Marsi, 2006); Both tools had to be slightly modified to handle the noisy POS tags often predicted on web data.[2]

It is worth noting that in the above task description is only one of possibly many instantiations of domain adaptation for parsing the web. We believe that this setup is the most realistic. One could argue that it is overly restrictive. In order to construct parsers that obtain high accuracy across all web domains, the simplest solution might be to annotate a new set of diverse sentences from the web. This will help account for both lexical and structural divergence. However, such a solution might not be scalable as it is unlikely one can annotate examples from all text domains represented on the web. In fact, not only is classifying a web document into a set of predefined domains a hard prediction task in and of itself, simply defining the set of all domains on the web can be non-trivial. If we add to this the fact that our social and topic domain space changes frequently, then annotation efforts would need to be an ongoing process as opposed to a one-off cost. Having said that, it is likely that the head of the distribution can be adequately covered. In fact, the domains that make up the Google Web Treebank do already span a large portion of texts occurring on the web from which we may consider extracting information. However, in order to handle the elusive tail of the domain space and the ever growing set of possible domains, we believe further studies into adapting parsing technology is a necessity, which is what motivates the setup for this particular shared-task. In all likelihood, a combination of additional annotation and robust domain adaptation will be required to bring parser accuracies inline with those observed for edited texts.

4 Systems

The shared task received 20 official submissions from 11 institutions in total. Of these, there were 8 submissions in the constituency parsing track and 12 submissions in the dependency parsing track. Here we list each system by its official name as well as a citation to the complete system description. The 8 constituency submissions received were:

- Alpage-1 & 2 – Seddah et al. (2012)
- DCU-Paris13-1 & 2 – Le Roux et al. (2012)
- IMS – Bohnet et al. (2012)
- OHSU – Dunlop and Roark (2012)
- Stanford – McClosky et al. (2012)
- Vanderbilt – Tang et al. (2012)

and the 12 dependency submissions received were:

- CPH-Trento – Søgaard and Plank (2012)
- DCU-Paris13 – Le Roux et al. (2012)
- HIT-Baseline & Domain – Zhang et al. (2012)
- IMS-1, 2 & 3 – Bohnet et al. (2012)
- NAIST – Hayashi et al. (2012)
- Stanford-1 & 2 – McClosky et al. (2012)
- UMass – Wu and Smith (2012)
- UPenn – Pitler (2012)

For dependency parsing, in addition to the above 12 systems, we also created submissions for each of the 8 constituency parser submissions by running the output through the Stanford converter.

These submissions investigated a wide variety of techniques to tackle the problem. However, there were a few underlying commonalities that spanned multiple systems.

- Many systems built on top of publicly available tools, most frequently the Berkeley parser (Petrov et al., 2006), the Charniak parser (Charniak, 2000) and the Mate dependency parser (Bohnet, 2010).

- Combination systems were prevalent, in particular for the highest ranking systems. This includes product-of-experts (Alpage, DCU-Paris13), stacking (IMS, Stanford, UPenn), voting (CPH-Trento, DCU-Paris13, HIT), bagging (HIT), up-training (IMS), re-ranking (DCU-Paris13, IMS, Stanford) and model merging (OHSU, Stanford).

[2] The modified scripts are available at http://mlcomp.org

- Unlabeled data was used by many systems most commonly for self-training (Alpage, DCU-Paris13, IMS, OHSU, Stanford, Vanderbilt) and generating clusters or embeddings (Alpage, IMS, NAIST, UMASS, Vanderbilt), but also to aid techniques like co/tri-training (HIT, NAIST), bootstrapping (Alpage), instance weighting (CPH-Trento) and genre classification (DCU-Paris13).

- Many teams focused on improving the base part-of-speech tagger, in particular for dependency parsing systems where this is more commonly used as a pre-processor as opposed to being integrated in the search. The primary technique here was to use word cluster features, but stacking (HIT, Vanderbilt) and data preprocessing (Alpage, DCU-Paris13, Stanford) were also investigated.

- A few teams pre-processed the data, either normalizing and correcting particular web-related tokens or part-of-speech tags (Alpage, Stanford) or by augmenting the treebanks (IMS, UPenn).

On top of the official submissions we prepared two baselines, one for each track. These baselines were trained on the WSJ portion of the training data only and did not include any additional resources. For the constituency track we trained the publicly available Berkeley parser (Petrov et al., 2006) which produces both part-of-speech tags and constituency structure. We call this system BerkeleyParser. For dependency parsing we trained a reimplementation of the shift-reduce parsing model of Zhang and Nivre (2011) with a beam of 64. For part-of-speech tags we used the TnT tagger (Brants, 2000). We call this system ZhangNivre.

5 Results and Discussion

Results are given in Table 3 for the constituency parsing track and Table 4 for the dependency parsing track. It is immediately clear that domain adaptation for parsing the web is far from a solved problem. Though systems routinely have newswire accuracies in excess of 89% for constituency parsers and 90% for dependency parsers, the best reporting systems for the web data score in the 82-84% range for constituency parsers and 81-85% range for dependency parsers. Even part-of-speech tagging, which is often considered a solved problem, poses tremendous challenges with accuracies around 90%. The problem is most acute for the answers domain, which is furthest from WSJ, particularly in the kinds of syntactic structures that it contains (questions, imperatives, etc.). This suggests that when moving to even more distant domains we can expect further degradation in performance, e.g., for social media texts (Gimpel et al., 2011; Foster et al., 2011).

Another important observation is that parser combinations dominate the highest scoring systems. The top-5 systems in both tracks involve some kind of system combination, be it product-of-experts, voting, stacking, combinations-of-combinations or some other technique. Past studies on parser adaptation have also shown combination systems to be robust to domain shift (Sagae and Tsujii, 2007). However, it is still unclear whether these systems are simply just better parsers overall or a truly robust to domain shift. Understanding this is an interesting and fundamental area of future work.

The bottom of Table 4 shows the results for the constituency outputs converted to dependencies with the Stanford converter. Here we can see that these systems tend to do better on average than the systems in the dependency track. In fact, the highest scoring dependency track system was itself a combination of converted constituency parsers. There are numerous reasons for this. As has been observed before (Petrov et al., 2010), constituency parsers tend to be better at parsing out-of-domain data due to global constraints placed on inference via grammar productions near the top of the tree.

The automatic conversion to dependencies also can favor constituency parsers in at least two respects. First, this conversion is lossy, which means that dependency parsers cannot model more fine-grained attachment distinctions due to their overall flatter structure. Also, as noted earlier, the Stanford converter made numerous mistakes converting the web data from constituents to dependencies that it did not typically make on the newswire from WSJ. As a result, there was a spurious domain shift that parsers trained only on the raw dependencies could never recover. However, since this shift did not exist for the constituency data, then constituency parsers would not suffer from this problem. This is because the conversion process on the evaluation data naturally include the systematic errors and/or inconsistencies made by the Stanford converter leading to seemingly correct dependencies.

In general, better WSJ parsing led to better web parsing. Only a few systems displayed a low accuracy on WSJ compared to the web data when put in context of the other systems. This might suggest that the current best methodology for attacking the domain adaptation problem is to improve parsing on the WSJ, and not to directly address domain shift. However, we believe that this is an artifact of the plethora of combination systems submitted to the shared-task. So while focusing on parsing the WSJ could in theory lead to improvements out-of-domain, such a strategy will never completely bridge the gap in parser accuracy.

In fact, the top placed DCU-Paris13 team shows that performance on the WSJ is not necessarily a good indicator for how a accurate parser will be on web text. Their dependency parsing system achieved the best accuracy on the web text, but was only the 7^{th} best on the WSJ-eval data. In the constituency track, the second ranked DCU-Paris13-2 system had the 6^{th} best performance on the WSJ-eval data. Thus, studies focusing on improving parsing across a wide-range of domains are certainly the most valuable. Hopefully the resources made available through this shared task will aid in breaking the communities reliance on WSJ section 23 parsing results as the sole (or at least primary) means for measuring English parsing performance.

Finally, we observe that better tagging was also highly correlated with high parsing accuracy. This is particularly true for the dependency parsing track. Unlike constituency parsers, dependency parsers tend to use part-of-speech tags as input to the parser as opposed to part of the syntax that must be predicted jointly. As a result, they are highly dependent on the accuracy of such tags. Furthermore, many teams reported significant improvements in LAS when using gold tags. This suggests that tackling part-of-speech tagging accuracy on the web is a fruitful means for improving parsing on the web, at least for dependencies.

6 Conclusions

Domain adaptation for parsing the web is still an unsolved problem. Ultimately, the most successful systems were those that used combinations in order to improve their parsers across all domains. However, additional efforts were made to truly adapt parsers via self-training, cluster features, instance weighting and smart pre-processing. It is our hope that the data sets used in this shared-task[3] will spur further research in bridging the gap in accuracy between edit texts, such as newswire, and unedited texts that are more frequent on the web.

References

Bernd Bohnet, Richárd Farkas, and Özlem Çetinoğlu. 2012. Sancl 2012 shared task: The ims system description. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proc. of COLING*.

Thorsten Brants. 2000. TnT - a statistical part-of-speech tagger. In *Proc. of ANLP*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of CoNLL*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of NAACL*.

Marie-Catherine De Marneffe, Bill MacCartney, and Chris D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC*.

Aaron Dunlop and Brian Roark. 2012. Merging self-trained grammars for automatic domain adaptation. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Jennifer Foster, Özlem Çetinoğlu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. # hardtoparse: Pos tagging and parsing the twitterverse. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Kevin Gimpel, Nanthan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. of ACL*.

[3] It will be made available through the LDC.

Katsuhiko Hayashi, Shuhei Kondo, Kevin Duh, and Yuji Matsumoto. 2012. The NAIST dependency parser for SANCL2012 shared task. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

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 *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

David McClosky, Wanxiang Che, Marta Recasens, Mengqiu Wang, Richard Socher, and Christopher D. Manning. 2012. Stanford's System for Parsing the English Web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL*.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proc. of EMNLP*.

Emily Pitler. 2012. Conjunction representation and ease of domain adaptation. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *Proc. of CoNLL Shared Task*.

Djamé Seddah, Benoît Sagot, and Marie Candito. 2012. The Alpage Architecture at the SANCL 2012 Shared Task: Robust Preprocessing and Lexical bridging for user-generated content parsing. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

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

Buzhou Tang, Min Jiang, and Hua Xu. 2012. Varderbilt's system for sancl2012 shared task. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Xiaoye Wu and David A. Smith. 2012. Semi-supervised deterministic shift-reduce parsing with word embeddings. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proc. of ACL*.

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)*.

	Domain A (Answers)				Domain B (Newsgroups)				Domain C (Reviews)				Domain D (WSJ)				Average (A-C)			
Team	LP	LR	F1	POS	LP	LR	F1	POS	LP	LR	F1	POS	LP	LR	F1	POS	LP	LR	F1	POS
BerkeleyParser	75.86	75.98	75.92	90.20	77.87	78.42	78.14	91.24	77.65	76.68	77.16	89.33	88.34	88.08	88.21	97.08	77.13	77.03	77.07	90.26
OHSU	73.21	74.60	73.90	90.15	73.22	75.48	74.33	91.14	76.31	76.22	76.27	90.05	83.17	83.79	83.48	96.84	74.25	75.43	74.83	90.45
Vanderbilt	75.09	76.78	75.93	**91.76**	78.10	79.05	78.57	92.91	77.74	78.18	77.96	91.94	87.82	88.00	87.91	97.49	76.98	78.00	77.49	92.20
IMS	79.46	78.10	78.78	90.22	80.85	80.12	80.48	91.09	81.31	78.61	79.94	89.93	89.83	88.96	89.39	97.31	80.54	78.94	79.73	90.41
Stanford	78.79	77.91	78.35	91.21	81.41	80.49	80.95	91.62	81.95	80.32	81.13	92.45	90.00	88.93	89.46	97.01	80.72	79.57	80.14	91.76
Alpage-1	80.67	80.36	80.52	91.17	84.22	83.12	83.67	93.22	82.01	81.04	81.52	91.58	90.20	89.62	89.91	97.20	82.30	81.51	81.90	91.99
Alpage-2	80.77	80.43	80.60	91.14	84.71	83.36	84.03	92.58	82.28	81.24	81.76	91.63	90.19	89.56	89.87	97.22	82.59	81.68	82.13	91.78
DCU-Paris13-2	80.02	79.22	79.62	91.61	83.13	82.18	82.65	**93.60**	82.92	82.12	82.52	**92.96**	88.43	88.29	88.36	97.29	82.02	81.17	81.60	**92.72**
DCU-Paris13-1	**82.96**	**81.43**	**82.19**	91.63	**85.01**	**83.65**	**84.33**	93.39	**84.79**	**83.29**	**84.03**	92.89	**90.75**	**90.32**	**90.53**	**97.53**	**84.25**	**82.79**	**83.52**	92.64

Table 3: Constituency parsing results. BerkeleyParser is the baseline model trained only on newswire. Numbers in bold are the highest score for each metric. Note that the average is computed only across the web domains.

	Domain A (Answers)			Domain B (Newsgroups)			Domain C (Reviews)			Domain D (WSJ)			Average (A-C)		
Team	LAS	UAS	POS	LAS	UAS	POS	LAS	UAS	POS	LAS	UAS	POS	LAS	UAS	POS
ZhangNivre	76.60	81.59	89.74	81.62	85.19	91.17	78.10	83.32	89.60	89.37	91.46	96.84	78.77	83.37	90.17
UPenn	68.54	82.28	89.65	74.41	86.10	90.99	70.17	82.88	89.02	81.74	91.99	96.93	71.04	83.75	89.89
UMass	72.51	78.36	89.42	77.23	81.61	91.28	74.89	80.34	89.90	81.15	83.97	94.71	74.88	80.10	90.20
NAIST	73.54	79.89	89.92	79.83	84.59	91.39	75.72	81.99	90.47	87.95	90.99	97.40	76.36	82.16	90.59
IMS-2	74.43	80.77	89.50	79.63	84.29	90.72	76.55	82.18	89.41	86.88	89.90	97.02	76.87	82.41	89.88
IMS-3	75.90	81.30	88.24	79.77	83.96	89.70	77.61	82.38	88.15	86.02	88.89	95.14	77.76	82.55	88.70
IMS-1	78.33	83.20	91.07	83.16	86.86	91.70	79.02	83.82	90.01	90.82	92.73	97.57	80.17	84.63	90.93
CPH-Trento	78.12	82.91	90.42	82.90	86.59	91.15	79.58	84.13	89.83	90.47	92.42	97.25	80.20	84.54	90.47
Stanford-2	77.50	82.57	90.30	83.56	87.18	91.49	79.70	84.37	90.46	89.87	91.95	95.00	80.25	84.71	90.75
HIT-Baseline	80.75	85.84	90.99	85.26	88.90	92.32	81.60	86.60	90.65	**91.88**	**93.88**	**97.76**	82.54	87.11	91.32
HIT-Domain	80.79	**85.86**	90.99	85.18	88.81	92.32	81.92	86.80	90.65	91.82	93.83	**97.76**	82.63	87.16	91.32
Stanford-1	81.01	85.70	90.30	**85.85**	**89.10**	91.49	82.54	86.73	90.46	91.50	93.38	95.00	83.13	87.18	90.75
DCU-Paris13	**81.15**	85.80	**91.79**	85.38	88.74	**93.81**	**83.86**	**88.31**	**93.11**	89.67	91.79	97.29	**83.46**	**87.62**	**92.90**
BerkeleyParser	77.42	82.38	90.19	82.24	85.84	91.18	77.90	83.02	89.33	89.68	91.78	97.12	79.19	83.75	90.23
OHSU	76.50	81.65	90.13	79.78	83.71	91.13	78.47	83.71	90.04	86.56	89.13	96.85	78.25	83.02	90.43
Vanderbilt	77.80	82.91	91.76	81.96	85.47	92.91	79.05	83.96	91.94	89.37	91.43	97.49	79.60	84.11	92.20
IMS	79.77	84.46	90.21	82.59	86.09	91.08	80.11	84.47	89.94	90.27	92.28	97.32	80.82	85.01	90.41
Alpage-1	80.47	85.31	91.10	84.96	88.01	93.17	81.43	86.14	91.37	90.65	92.68	97.20	82.29	86.49	91.88
Alpage-2	80.60	85.38	91.08	85.08	88.26	92.51	81.60	86.28	91.41	90.52	92.55	97.22	82.43	86.64	91.67
Stanford	81.13	85.80	91.20	84.05	87.51	91.61	83.22	87.26	92.45	90.28	92.53	97.01	82.80	86.86	91.75
DCU-Paris13-2	80.47	85.30	91.60	84.82	88.27	93.58	83.70	88.17	92.96	89.67	91.79	97.29	83.00	87.90	92.71
DCU-Paris13-1	81.71	86.49	91.62	85.47	88.78	93.38	84.19	88.44	92.89	91.37	93.33	97.53	83.79	87.90	92.63

Table 4: Dependency parsing results. Upper half are official submissions to the dependency parsing track. Lower half are the systems from the constituency track converted to dependencies using the Stanford Converter. ZhangNivre and BerkeleyParser are the baseline models trained only on newswire. Numbers in bold in the top half of the table represent the highest score for each metric. Underlined numbers in the bottom half of the table represent what would have been the highest score for each metric had the constituency to dependency conversion been submitted as an official result. The average is computed only across the web domains.