"cba to check the spelling" Investigating Parser Performance on Discussion Forum Posts

Jennifer Foster
National Centre for Language Technology
School of Computing
Dublin City University
jfoster@computing.dcu.ie

Abstract

We evaluate the Berkeley parser on text from an online discussion forum. We evaluate the parser output with and without gold tokens and spellings (using Sparseval and Parseval), and we compile a list of problematic phenomena for this domain. The Parseval f-score for a small development set is 77.56. This increases to 80.27 when we apply a set of simple transformations to the input sentences and to the Wall Street Journal (WSJ) training sections.

1 Introduction

Parsing techniques have recently become efficient enough for parsers to be used as part of a pipeline in a variety of tasks. Another recent development is the rise of user-generated content in the form of blogs, wikis and discussion forums. Thus, it is both interesting and necessary to investigate the performance of NLP tools trained on edited text when applied to unedited Web 2.0 text. McClosky et al. (2006) report a Parseval f-score decrease of 5% when a WSJ-trained parser is applied to Brown corpus sentences. In this paper, we move even further from the WSJ by investigating the performance of the Berkeley parser (Petrov et al., 2006) on user-generated content.

We create gold standard phrase structure trees for the posts on two threads of the same online discussion forum. We then parse the sentences in one thread, our development set, with the Berkeley parser under three conditions: 1) when it performs its own tokenisation, 2) when it is provided with gold tokens and 3) when misspellings in the input have been corrected. A qualitative evaluation is then carried out on parser output under the third condition. Based on this evaluation, we identify some "low-hanging fruit" which we attempt to handle either by transforming the input sentence or by transforming the WSJ training material. The success of these transformations is evaluated on our development and test sets, with encouraging results.

2 Parser Evaluation Experiments

Data Our data consists of sentences that occur on the BBC Sport 606 Football discussion forum. The posts on this forum are quite varied, ranging from throwaway comments to more considered essay-like contributions. The development set consists of 42 posts (185 sentences) on a thread discussing a controversial refereeing decision in a soccer match.[1] The test set is made up of 40 posts (170 sentences) on a thread discussing a player's behaviour in the same match.[2] The average sentence length in the development set is 18 words and the test set 15 words. Tokenisation and spelling correction were carried out by hand on the sentences in both sets.[3] They were then parsed using Bikel's parser (Bikel, 2004) and corrected by hand using the Penn Treebank Bracketing Guidelines (Bies et al., 1995).

Parser The Berkeley parser is an unlexicalised phrase structure parser which learns a latent variable PCFG by iteratively splitting the treebank non-

[1] http://www.bbc.co.uk/dna/606/F15264075?thread=7065503&show=50

[2] http://www.bbc.co.uk/dna/606/F15265997?thread=7066196&show=50

[3] Note that abbreviated forms such as *cos* which are typical of computer-mediated communication are not corrected.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 381–384,
Los Angeles, California, June 2010.

terminals, estimating rule probabilities for the new grammar using EM and merging the less useful splits. We train a PCFG from WSJ2-21 by carrying out five cycles of the split-merge process (SM5).

Tokenisation and Spelling Effects In the first experiment, the parser is given the original development set sentences which contain spelling mistakes and which have not been tokenised. We ask the parser to perform its own tokenisation. In the second experiment, the parser is given the hand-tokenised sentences which still contain spelling mistakes. These are corrected for the third experiment.

Since the yields of the parser output and gold trees are not guaranteed to match exactly, we cannot use the `evalb` implementation of the Parseval evaluation metrics. Instead we use Sparseval (Roark et al., 2006), which was designed to be used to evaluate the parsing of *spoken* data and can handle this situation. An unaligned dependency evaluation is carried out: head-finding rules are used to convert a phrase structure tree into a dependency graph. Precision and recall are calculated over the dependencies

The Sparseval results are shown in Table 1. For the purposes of comparison, the WSJ23 performance is displayed in the top row. We can see that performance suffers when the parser performs its own tokenisation. A reason for this is the under-use of apostrophes in the forum data, with the result that words such as *didnt* and *im* remain untokenised and are tagged by the parser as common nouns:
*(NP (NP (DT the) (NNS refs)) (SBAR (S (NP (NN **didnt**)) (VP want to make it to obvious))))*

To properly see the effect of the 39 spelling errors on parsing accuracy, we factor out the mismatches between the correctly spelled words in the reference set and their incorrectly spelled equivalents. We do this by evaluating against a version of the gold standard which contains the original misspellings (third row). We can see that the effect of spelling errors is quite small. The Berkeley parser's mechanism for handling unknown words makes use of suffix information and it is able to ignore many of the content word spelling errors. It is the errors in function words that appear to cause a greater problem:
*(NP (DT the) (JJ zealous) (NNS fans) **(NN whpo)** (NN care) (JJR more))*

Test Set	R	P	F
WSJ 23	88.66	88.66	88.66
Football	68.49	70.74	69.60
Football Gold Tokens	71.54	73.25	72.39
Ft Gold Tok (misspelled gold)	73.49	75.25	74.36
Football Gold Tokens+Spell	73.94	75.59	74.76

Table 1: **Sparseval** scores for Berkeley SM5

Test Set	R	P	F
WSJ 23	88.88	89.46	89.17
Football Gold Tokens+Spell	78.15	76.97	77.56

Table 2: **Parseval** scores for Berkeley SM5

Gold Tokens and Spelling Leaving aside the problems of automatic tokenisation and spelling correction, we focus on the results of the third experiment. The Parseval results are given in Table 2. Note that the performance degradation is quite large, more than has been reported for the Charniak parser on the Brown corpus. We examine the parser output for each sentence in the development set. The phenomena which lead the parser astray are listed in Table 3. One problem is coordination which is difficult for parsers on in-domain data but which is exacerbated here by the omission of conjunctions, the use of a comma as a conjunction and the tendency towards unlike constituent coordination.

Parser Comparison We test the lexicalised Charniak parser plus reranker (Charniak and Johnson, 2005) on the development set sentences. We also test the Berkeley parser with an SM6 grammar. The f-scores are shown in Table 4. The parser achieving the highest score on WSJ23, namely, the C&J reranking parser, also achieves the highest score on our development set. The difference between the two Berkeley grammars supports the claim that an SM6 grammar overfits to the WSJ (Petrov and Klein, 2007). However, the differences between the four parser/grammar configurations are small.

Parser	WSJ23	Football
Berkeley SM5	89.17	77.56
Berkeley SM6	89.56	77.01
Charniak First-Stage	89.13	77.13
C & J Reranking	91.33	78.33

Table 4: Cross-parser and cross-grammar comparison

Problematic Phenomena	Examples
Idioms/Fixed Expressions	*Spot on* (S (VP (VB Spot) (PP (IN on))) (. .))
Acronyms	*lmao* (S (NP (PRP you)) (VP (VBZ have) (RB n't) (VP (VBN done) (NP (ADVP (RB that) (RB once)) (DT this) (NN season)) (NP (NN lmao)))))
Missing subject	*Does n't change the result though* (SQ (VBZ Does) (RB n't) (NP (NN change)) (NP (DT the) (NN result)) (ADVP (RB though)) (. !))
Lowercase proper nouns	*paul scholes* (NP (JJ paul) (NNS scholes))
Coordination	*Very even game and it's sad that...* (S (ADVP (RB Very)) (NP (NP (JJ even) (NN game)) (CC and) (NP (PRP it))) (VP (VBZ 's) (ADJP (JJ sad)) (SBAR (IN that)...
Adverb/Adjective Confusion	*when playing bad* (SBAR (WHADVP (WRB when)) (S (VP (VBG playing) (ADJP (JJ bad)))))
CAPS LOCK IS ON	*YOU GOT BEATEN BY THE BETTER TEAM* (S (NP (PRP YOU)) (VP (VBP GOT) (NP (NNP BEATEN) (NNP BY) (NNP THE) (NNP BETTER) (NNP TEAM))))
cos instead of *because*	*or it was cos you lost* (VP (VBD was) (ADJP (NN cos) (SBAR (S (NP (PRP you)) (VP (VBD lost))))))

Table 3: Phenomena which lead the parser astray. The output of the parser is given for each example.

3 Initial Improvements

Parsing performance on noisy data can be improved by transforming the input data so that it resembles the parser's training data (Aw et al., 2006), transforming the training data so that it resembles the input data (van der Plas et al., 2009), applying semi-supervised techniques such as the self-training protocol used by McClosky et al. (2006), and changing the parser internals, e.g. adapting the parser's unknown word model to take into account variation in capitalisation and function word misspelling.[4]

We focus on the first two approaches and attempt to transform both the input data and the WSJ training material. The transformations that we experiment with are shown in Table 5. The treebank transformations are performed in such a way that their frequency distribution mirrors their distribution in the development data. We remove discourse-marking acronyms such as *lol*[5] from the input sentence, but do not attempt to handle acronyms which are integrated into the sentence.[6]

We examine the effect of each transformation on development set parsing performance and discard those which do not improve performance. We keep all the input sentence transformations and those treebank transformations which affect lexical rules, i.e. changing the endings on adverbs and changing the first character of proper nouns. The treebank transformations which delete subject pronouns and coordinating conjunctions are not as effective. They work in individual cases, e.g. the original analysis of the sentence *Will be here all day* is
(S (NP (NNP Will)) (VP be here all day) (. .))
After applying the treebank transformation, it is
(S (VP (MD Will) (VP be here all day)) (. .))
Their overall effect is, however, negative. It is likely that, for complex phenomena such as coordination and subject ellipsis, the development set is still too small to inform how much of and in what way the original treebank should be transformed. The results of applying the effective transformations to the development set and the test set are shown in Table 6.

[4]Even when spelling errors have been corrected, unknown words are still an issue: 8.5% of the words in the football development set do not occur in WSJ2-21, compared to 3.6% of the words in WSJ23.

[5]In a study of teenage instant messaging, Tagliamonte and Dennis (2008) found that forms such as *lol* are not as ubiquitous as is commonly perceived. Although only occurring a couple of times in our data, they are problematic for the parser.

[6]An example is: *your loss to Wigan would be more scrutunized (**cba** to check spelling) than it has been this year*

Input Sentence
cos → *because*
Sentences consisting of all uppercase characters converted to standard capitalisation *DEAL WITH IT* → *Deal with it*
Remove certain acronyms *lol* → ϵ
Treebank
Delete subject noun phrases when the subject is a pronoun *(S (NP (PRP It)) (VP (VBD arrived)...* ⟶ *(S (VP (VBD arrived)...*
Delete or replace conjunctions with a comma (for sentence coordination) *(S ...) (CC and) (S ...)* ⟶ *(S ...) (, ,) (S ...)* OR *(S ...) (CC and) (S ...)* ⟶ *(S ...) (S ...)*
Delete *ly* from adverbs *(VP (VBD arrived) (ADVP (RB quickly)))* ⟶ *(VP (VBD arrived) (ADVP (RB quick)))*
Replace uppercase first character in proper nouns *(NP (NP (NNP Warner) (POS 's)) (NN price))* ⟶ *(NP (NP (NNP warner) (POS 's)) (NN price))*

Table 5: Input Sentence and Treebank Transformations

Configuration	Recall	Precision	F-Score
Baseline Dev	78.15	76.97	77.56
Transformed Dev	80.83	79.73	80.27
Baseline Test	77.61	79.14	78.37
Transformed Test	80.10	79.77	79.93

Table 6: Effect of transformations on dev and test set

The recall and precision improvements on the development set are statistically significant ($p < 0.02$), as is the recall improvement on the test set ($p < 0.05$).

4 Conclusion

Ongoing research on the problem of parsing unedited informal text has been presented. At the moment, because of the small size of the data sets and the variety of writing styles in the development set, only tentative conclusions can be drawn. However, even this small data set reveals clear problems for WSJ-trained parsers: the handling of long coordinated sentences (particularly in the presence of erratic punctuation usage), domain-specific fixed expressions and unknown words. We have presented some preliminary experimental results using simple transformations to both the input sentence and the parser's training material. Treebank transformations need to be more thoroughly explored with use made of the Switchboard corpus as well as the WSJ.

Acknowledgments

Thanks to the reviewers and to Emmet Ó Briain, Deirdre Hogan, Adam Bermingham, Joel Tetreault.

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalisation. In *Proceedings of the 21st COLING/44th ACL*.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style, Penn Treebank Project. Technical Report Tech Report MS-CIS-95-06, University of Pennsylvania.

Daniel Bikel. 2004. Intricacies of Collins Parsing Model. *Computational Linguistics*, 30(4):479–511.

Eugene Charniak and Mark Johnson. 2005. Course-to-fine n-best-parsing and maxent discriminative reranking. In *Proceedings of the 43rd ACL*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st COLING/44th ACL*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of HLT NAACL 2007*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact and interpretable tree annotation. In *Proceedings of the 21st COLING and the 44th ACL*.

Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr, Mark Johnson, Jeremy G. Kahn, Yang Liu, Mari Ostendorf, John Hale, Anna Krasnyanskaya, Matthew Lease, Izhak Shafran, Matthew Snover, Robin Stewart, and Lisa Yung. 2006. SParseval: Evaluation metrics for parsing speech. In *Proceedings of LREC*.

Sali A. Tagliamonte and Derek Dennis. 2008. Linguistic ruin? LOL! Instant messaging and teen language. *American Speech*, 83(1).

Lonneke van der Plas, James Henderson, and Paola Merlo. 2009. Domain adaptation with artificial data for semantic parsing of speech. In *Proceedings of HLT NAACL 2009, Companion Volume: Short Papers*.