In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 145–147, Lisbon, Portugal, 2000.

Shallow Parsing as Part-of-Speech Tagging*

Miles Osborne
University of Edinburgh
Division of Informatics
2 Buccleuch Place
Edinburgh EH8 9LW , Scotland
osborne@cogsci.ed.ac.uk

Abstract

Treating shallow parsing as part-of-speech tagging yields results comparable with other, more elaborate approaches. Using the CoNLL 2000 training and testing material, our best model had an accuracy of 94.88%, with an overall FB1 score of 91.94%. The individual FB1 scores for NPs were 92.19%, VPs 92.70% and PPs 96.69%.

1 Introduction

Shallow parsing has received a reasonable amount of attention in the last few years (for example (Ramshaw and Marcus, 1995)). In this paper, instead of modifying some existing technique, or else proposing an entirely new approach, we decided to build a shallow parser using an off-the-shelf part-of-speech (POS) tagger. We deliberately did not modify the POS tagger's internal operation in any way. Our results suggested that achieving reasonable shallow-parsing performance does not in general require anything more elaborate than a simple POS tagger. However, an error analysis suggested the existence of a small set of constructs that are not so easily characterised by finite-state approaches such as ours.

2 The Tagger

We used Ratnaparkhi's maximum entropy-based POS tagger (Ratnaparkhi, 1996). When tagging, the model tries to recover the most likely (unobserved) tag sequence, given a sequence of observed words.

For our experiments, we used the binary-only distribution of the tagger (Ratnaparkhi, 1996).

* The full version of this paper can be found at http://www.cogsci.ed.ac.uk/~osborne/shallow.ps

3 Convincing the Tagger to Shallow Parse

The insight here is that one can view (some of) the differences between tagging and (shallow) parsing as one of context: shallow parsing requires access to a greater part of the surrounding lexical/POS syntactic environment than does simple POS tagging. This extra information can be encoded in a *state*.

However, one must balance this approach with the fact that as the amount of information in a state increases, with limited training material, the chance of seeing such a state again in the future diminishes. We therefore would expect performance to increase as we increased the amount of information in a state, and then decrease when overfitting and/or sparse statistics become dominate factors.

We trained the tagger using 'words' that were various 'configurations' (concatenations) of actual words, POS tags, chunk-types, and/or suffixes or prefixes of words and/or chunk-types. By training upon these concatenations, we help bridge the gap between simple POS tagging and shallow parsing.

In the rest of the paper, we refer to what the tagger considers to be a word as a *configuration*. A configuration will be a concatenation of various elements of the training set relevant to decision making regarding chunk assignment. A 'word' will mean a word as found in the training set. 'Tags' refer to the POS tags found in the training set. Again, such tags may be part of a configuration. We refer to what the tagger considers as a tag as a *prediction*. Predictions will be chunk labels.

4 Experiments

We now give details of the experiments we ran. To make matters clearer, consider the following fragment of the training set:

Word	w_1	w_2	w_3
POS Tag	t_1	t_2	t_3
Chunk	c_1	c_2	c_3

Words are w_1, w_2 and w_3, tags are t_1, t_2 and t_3 and chunk labels are c_1, c_2 and c_3. Throughout, we built various configurations when predicting the chunk label for word w_1.[1]

With respect to the situation just mentioned (predicting the label for word w_1), we gradually increased the amount of information in each configuration as follows:

1. A configuration consisting of just words (word w_1). Results:

Chunk type	P	R	FB1
Overall	88.06	88.71	88.38
ADJP	67.57	51.37	58.37
ADVP	74.34	74.25	74.29
CONJP	54.55	66.67	60.00
INTJ	100.00	50.00	66.67
LST	0.00	0.00	0.00
NP	87.84	89.41	88.62
PP	94.80	95.91	95.35
PRT	71.00	66.98	68.93
SBAR	82.30	72.15	76.89
VP	86.68	88.15	87.41

Overall accuracy: 92.76%

2. A configuration consisting of just tags (tag t_1). Results:

Chunk type	P	R	FB1
Overall	88.15	88.07	88.11
ADJP	67.99	54.79	60.68
ADVP	71.61	70.79	71.20
CONJP	35.71	55.56	43.48
INTJ	0.00	0.00	0.00
LST	0.00	0.00	0.00
NP	89.47	89.57	89.52
PP	87.70	95.28	91.33
PRT	52.27	21.70	30.67
SBAR	83.92	31.21	45.50
VP	90.38	91.18	90.78

Overall accuracy 92.66%.

3. Both words, tags and the current chunk label (w_1, t_1, c_1) in a configuration. We allowed the tagger access to the current chunk label by training *another* model with configurations consisting of tags and words (w_1 and t_1). The training set was then reduced to consist of just tag-word configurations and tagged using this model. Afterwards, we collected the predictions for use in the second model. Results:

Chunk type	P	R	FB1
Overall	89.79	90.70	90.24
ADJP	69.61	57.53	63.00
ADVP	74.72	77.14	75.91
CONJP	54.55	66.67	60.00
INTJ	50.00	50.00	50.00
LST	0.00	0.00	0.00
NP	89.80	91.12	90.4
PP	95.15	96.26	95.70
PRT	71.84	69.81	70.81
SBAR	85.63	80.19	82.82
VP	89.54	91.31	90.41

Overall accuracy: 93.79%

4. The final configuration made an attempt to take deal with sparse statistics. It consisted of the current tag t_1, the next tag t_2, the current chunk label c_1, the last two letters of the next chunk label c_2, the first two letters of the current word w_1 and the last four letters of the current word w_1. This configuration was the result of numerous experiments and gave the best overall performance. The results can be found in Table 1.

We remark upon our experiments in the comments section.

5 Error Analysis

We examined the performance of our final model with respect to the testing material and found that errors made by our shallow parser could be grouped into three categories: difficult syntactic constructs, mistakes made in the training or testing material by the annotators, and errors peculiar to our approach.[2]

Taking each category of the three in turn, problematic constructs included: co-ordination, punctuation, treating ditransitive VPs as being transitive VPs, confusions regarding adjective or adverbial phrases, and copulars seen as being possessives.

[1] For space reasons, we had to remove many of these experiments. The longer version of the paper gives relevant details.

[2] The raw results can be found at: http://www.cogsci.ed.ac.uk/~osborne/conll00-results.txt The mis-analysed sentences can be found at: http://www.cogsci.ed.ac.uk/~osborne/conll00-results.txt.

Mistakes (noise) in the training and testing material were mainly POS tagging errors. An additional source of errors were odd annotation decisions.

The final source of errors were peculiar to our system. Exponential distributions (as used by our tagger) assign a non-zero probability to all possible events. This means that the tagger will at times assign chunk labels that are illegal, for example assigning a word the label I-NP when the word is not in a NP. Although these errors were infrequent, eliminating them would require 'opening-up' the tagger and rejecting illegal hypothesised chunk labels from consideration.

6 Comments

As was argued in the introduction, increasing the size of the context produces better results, and such performance is bounded by issues such as sparse statistics. Our experiments suggest that this was indeed true.

We make no claims about the generality of our modelling. Clearly it is specific to the tagger used.

In more detail, we found that:

- PPs seem easy to identify.
- ADJP and ADVP chunks were hard to identify correctly. We suspect that improvements here require greater syntactic information than just base-phrases.
- Our performance at NPs should be improved-upon. In terms of modelling, we did not treat any chunk differently from any other chunk. We also did not treat any words differently from any other words.
- The performance using just words and just POS tags were roughly equivalent. However, the performance using both sources was better than when using either source of information in isolation. The reason for this is that words and POS tags have different properties, and that together, the specificity of words can overcome the coarseness of tags, whilst the abundance of tags can deal with the sparseness of words.

Our results were not wildly worse than those reported by Buchholz *et al* (Sabine Buchholz and Daelemans, 1999). This comparable level of performance suggests that shallow parsing (base phrasal recognition) is a fairly easy task. Improvements might come from better modelling, dealing with illegal chunk sequences, allowing multiple chunks with confidence intervals, system combination etc, but we feel that such improvements will be small. Given this, we believe that base-phrasal chunking is close to being a solved problem.

test data	precision	recall	$F_{\beta=1}$
ADJP	72.42%	64.16%	68.04
ADVP	75.94%	79.10%	77.49
CONJP	50.00%	55.56%	52.63
INTJ	100.00%	50.00%	66.67
LST	0.00%	0.00%	0.00
NP	91.92%	92.45%	92.19
PP	95.95%	97.44%	96.69
PRT	73.33%	72.64%	72.99
SBAR	86.40%	80.75%	83.48
VP	92.13%	93.28%	92.70
all	91.65%	92.23%	91.94

Table 1: The results for configuration 4. Overall accuracy: 94.88%

Acknowledgements

We would like to thank Erik Tjong Kim Sang for supplying the evaluation code, and Donnla Nic Gearailt for dictating over the telephone, and from the top-of-her-head, a Perl program to help extract wrongly labelled sentences from the results.

References

Claire Cardie and David Pierce. 1998. Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics*, pages 218–224.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text Chunking Using Transformation-Based Learning. In *Proceedings of the* 3^{rd} *ACL Workshop on Very Large Corpora*, pages 82–94, June.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Part-Of-Speech Tagger. In *Proceedings of Empirical Methods in Natural Language*, University of Pennsylvania, May. Tagger: `ftp://ftp.cis.upenn.edu/pub/adwait/jmx`.

Jorn Veenstra Sabine Buchholz and Walter Daelemans. 1999. Cascaded Grammatical Relation Assignment. In *Proceedings of EMNLP/VLC-99*. Association for Computational Linguistics.