Chapter 5: The Superparser

Eric Brill, Barbora Hladká

There has been a great deal of recent interest in classifier combination as a means for increasing classifier accuracy. We have already demonstrated (BrillACL98) that combining part of speech taggers can result in a significant accuracy improvement over any single tagger. Our group intended to develop multiple Czech parsers, therefore we planned to explore methods for combining these parsers.

Since we initially only had one successful parser, an adaptation of the Collins parser, we instead began by exploring methods for diversifying and combining a single parser. One of the most successful methods for doing this is known as boosting. Essentially, multiple learning iterations are carried out, where the classifier is trained, then applied to the training set. At each iteration, instances that are incorrectly classified are given a greater weight in the next iteration. By doing so, in each iteration the learner is forced to concentrate on instances it was unable to correctly classify in earlier iterations. In the end, all of the trained classifiers are combined via weighted voting.

The problem with applying boosting to the Collins parser is that the parser is very fast to train, but very slow to apply. Each iteration of boosting requires one training run and one application run. But applying the parser to a sufficiently large training set would take days, and hundreds of iterations are often required. Another method for diversifying a single classifier is bagging. In bagging, we get a family of classifiers by training on different portions of the training set. The method works as follows. We first create N training bags. A single training bag is obtained by taking a training set of size S and sampling this training set S times with replacement. Some training instances will occur multiple times in a bag, while others may not appear at all. Next, each bag is used to train a classifier. We now have N classifiers. These classifiers are then combined by simple voting.

Bagging is effective in cases where small variations in the training data result in significant differences in the resulting classifier. If training is relatively insensitive to small training data differences, then the N resulting classifiers will not be significantly different, and therefore combining these classifiers will not give any significant improvement over a single classifier.

For our first experiments on Czech, we took the Collins parser, which had been retrained for Czech. We took the Czech corpus training set and generated three training bags, which were then used to train three Collins parsers, one from each bag. Below we show the constituent accuracy as a function of the number of parsers that posited that constituent:

Number of Parsers	**Accuracy**
1	28.0
2	50.3
3	92.2

From this we can see that voting has promise, as the number of parsers agreeing on a constituent gives us a great deal of information as to the probability of that constituent being correct. Given this insight, we next went ahead and generated bagged parsers, combining their outputs as described above. Below

are the results for these experiments:

	Precision	**Recall**	**P+R**
Original Parser	77.0	77.0	153.9
1 Bag	76.0	76.0	151.9
6 Bags	81.1	74.6	155.9
10	80.7	75.4	156.2
16	80.6	76.0	156.6

First, we note that a bagged parser by itself underperforms the original parser. This is because the original parser is trained on a "true" distribution of parse trees, whereas a bagged parser has a different sentence distribution. Also, the bagged parser was not exposed to all of the sentences in the training set. However, we see that when we combine a set of bags, we achieve performance significantly better than the original parser.

While there are many applications for which an imbalance of precision and recall is not problematic, there are also cases when we want a balanced parse, in other words a dependency structure that is complete, with every word having one and only one link pointing to it. To allow for such a structure, we changed our voting scheme. Instead of voting on a constituent basis, the parsers voted for every word W which link X --> W should be chosen.

	Precision	**Recall**	**P+R**
Original	77.0	77.0	153.9
16 bags unbalanced	80.6	76.0	156.6
16 bags balanced	77.8	77.8	155.6

We see that this method of combination, resulting in a balanced precision and recall, still gives us an improvement over the original parser, but not as great an improvement (in terms of P+R) as was achieved when voting was done on a constituent basis.

At the end of the workshop we had a number of parsers at our disposal for Czech, so we next explored methods for combining the outputs of those parsers. Below we show the oracle accuracy for various combinations of parsers, in other words the accuracy when an oracle can choose constituents from the union of all constituents posited by at least one parser.

	Oracle Accuracy
Original	77.0
+ 10 Bags	86.8
+ 18 Bags	88.1
+ 18 + Oren + Dan	91.7
+ Oren + Dan	86.6

Each parser does seem to add potential information. However, note that the Oren and Dan parser together give fewer good constituents than can simply be realized via bagging the original Collins parser.

Below are the results for constituent voting, where each parser has a weighted vote. The vote is weighted by the parser's estimated accuracy.

	P+R
18 Bags	156.6
18 Bags + Original	156.6
18 + Orig + Dan	156.6
18 + Orig + Oren	156.6
18 + Orig + Oren + Dan	157.2

From this we can see that at least using this simple combination scheme, the Oren and Dan parser do not provide us with useful information beyond what can be exploited from the Collins parser via bagging.

Below we show our final results. The method we chose for our final evaluation was to combine the 21 bags, without using the other parsers or the original parser.

	DeV			**EvaL**		
	P	**R**	**P+R**	**P**	**R**	**P+R**
Baseline	77.0	77.0	153.9	79.1	79.1	158.3
Unrestricted	80.6	76.0	156.6	81.8	79.1	160.9
Restricted	77.8	77.8	155.6	79.9	79.9	159.8

We also ran experiments on English, by bagging the Collins English parser.

	P	**R**	**P+R**
Original Parser	88.7	88.4	177.1
1 Bag	87.6	87.6	175.2
Original + 3 bags	90.5	87.1	177.6

References

Eric Brill and Jun Wu(1998). Classifiers Combination for Improved Lexical Disambiguation (COLING/ACL 1998).

Thomas G. Dietterich(1997). Machine-Learning Research: Four Current Directions. AI Magazine. Winter 1997, pp97-136.