In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 157–159, Lisbon, Portugal, 2000.

Single-Classifier Memory-Based Phrase Chunking

Jorn Veenstra and **Antal van den Bosch**
ILK / Computational Linguistics, Tilburg University
{veenstra, antalb}@kub.nl

1 Introduction

In the shared task for CoNLL-2000, words and tags form the basic multi-valued features for predicting a rich phrase segmentation code. While the tag features, containing WSJ part-of-speech tags (Marcus et al., 1993), have about 45 values, the word features have more than 10,000 values. In our study we have looked at how memory-based learning, as implemented in the TiMBL software system (Daelemans et al., 2000), can handle such features. We have limited our search to single classifiers, thereby explicitly ignoring the possibility to build a meta-learning classifier architecture that could be expected to improve accuracy. Given this restriction we have explored the following:

1. The generalization accuracy of TiMBL with default settings (multi-valued features, overlap metric, feature weighting).
2. The usage of MVDM (Stanfill and Waltz, 1986; Cost and Salzberg, 1993) (Section 2), which should work well on word value pairs with a medium or high frequency, but may work badly on word value pairs with low frequency.
3. The straightforward unpacking of feature values into binary features. On some tasks we have found that splitting multi-valued features into several binary features can enhance performance of the classifier.
4. A heuristic search for complex features on the basis of all unpacked feature values, and using these complex features for the classification task.

2 Methods and Data

The data used for this shared task is comparable to the dataset used in (Buchholz et al., 1999), who found an optimal windowing context size of five words and POS tags to the left, the word itself, and three words and POS tags to the right. We also used this window size, and have applied TiMBL to the shared task data using default TiMBL settings. TiMBL and the abovementioned feature metrics are introduced in the following sections.

IB1-IG The default TiMBL setting, IB1-IG, (Daelemans et al., 1997) is a memory-based learning algorithm that builds a database of instances (the *instance base*) during learning. An instance consists of a fixed-length vector of n feature-value pairs, and an information field containing the classification of that particular feature-value vector. After the instance base is built, new (test) instances are classified by matching them to all instances in the instance base, and by calculating with each match the *distance* between the new instance X and the memory instance Y.

The most basic metric for patterns with symbolic features is the **Overlap metric** given in equation 1; where $\Delta(X,Y)$ is the distance between patterns X and Y, represented by n features, w_i is a weight for feature i, and δ is the distance per feature. The k-NN algorithm with this metric, and equal weighting for all features is called IB1 (Aha et al., 1991). Usually k is set to 1.

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \ \delta(x_i, y_i) \qquad (1)$$

where: $\delta(x_i, y_i) = 0 \ if \ x_i = y_i, \ else \ 1$

This distance metric simply counts the number of (mis)matching feature values in both pat-

method	k	AdjP	AdvP	ConjP	Intj	NP	PP	PRT	SBAR	VP	tot
IB1-IG	1	60.9	75.3	17.6	66.7	91.0	95.9	65.4	78.2	91.6	90.5
IB1-IG	3	64.3	76.8	38.5	66.7	91.5	95.8	61.4	79.6	91.7	91.0
IB1-IG	5	65.4	76.5	38.5	66.7	91.6	95.8	63.7	78.6	91.6	91.0
IB1-IG	7	66.1	76.5	41.7	66.7	91.2	95.4	63.7	76.8	91.4	90.7
MVDM-all	1	58.3	73.9	34.5	0	90.2	95.6	54.6	78.1	89.7	89.6
MVDM-all	3	60.0	77.0	30.8	0	91.0	96.2	60.6	80.0	91.6	90.9
MVDM-all	5	61.0	76.8	30.8	0	91.3	96.1	60.1	79.5	91.9	91.0
MVDM-all	7	62.4	76.5	40.0	0	91.4	96.0	59.8	78.8	91.9	91.1
MVDM-POS	1	59.3	76.6	12.5	50.0	89.6	96.0	69.5	78.3	91.1	89.8
MVDM-POS	3	65.9	77.4	26.7	66.7	91.8	96.6	74.1	81.6	92.3	**91.5**
MVDM-POS	5	63.7	76.6	37.0	66.7	92.1	96.4	71.4	79.8	92.1	91.5
MVDM-POS	7	65.2	77.2	41.7	66.7	92.0	96.3	70.7	79.6	92.0	91.5
Unpacked features	1	49.4	72.9	47.1	0	88.7	95.8	59.1	79.3	89.1	88.8
Complex features	1	58.8	75.8	0	66.7	91.0	94.7	74.6	87.8	94.3	91.3

Table 1: Results on the shared task dataset, in the top row the best performing metric is shown.

terns. In the absence of information about feature relevance, this is a reasonable choice. However, Information Theory gives us a useful tool for measuring feature relevance (Quinlan, 1986; Quinlan, 1993). **Information Gain** (IG) weighting looks at each feature in isolation, and measures how much information it contributes to our knowledge of the correct class label. The Information Gain of feature f is measured by computing the difference in uncertainty (i.e. entropy) between the situations without and with knowledge of the value of that feature. The resulting IG values can then be used as weights in equation 1.

Modified Value Difference Metric The Modified Value Difference Metric (MVDM) (Cost and Salzberg, 1993) estimates the distance between two values of a feature by comparing the class distribution of both features. MVDM can give good estimates if there are enough occurrences of the two values, but for low-frequent values unreliable values of MVDM can occur. For this data we can expect that this sparseness effect hinders the word features more than the POS features.

Unpacking Features Unpacking features implies that all feature values receive individual weights. (Van den Bosch and Zavrel, 2000) warn that this operation forces feature weights to be based on less observations, which could make the weights unrealistic in view of test data. Moreover, the k nearest neighbors can be expected to contain less instances with a fixed k when unpacking features; this is usually not beneficial for generalization accuracy (Van den Bosch and Zavrel, 2000).

Complex Features In the previously discussed versions of memory-based learning, features are treated as independent. However, sometimes combinations of features or feature values may be very good predictors. Since there are many possible combinations, search strategies are needed to select the best. Such strategies have been developed for rule induction algorithms (Clark and Niblett, 1989; Quinlan, 1993; Cohen, 1995), and they can be used to find complex features for memory-based learning as well. We followed the following procedure:

1. apply Ripper (Cohen, 1995) to the training set, and collect the set of induced rules;
2. recode the instances in the training and test set, by setting binary features denoting the rules that apply to them;
3. apply memory-based learning to the recoded training set, and classify the recoded test set.

3 Experiments and Results

In Table 2 we give an overview of the experiments with different metrics and settings. In the first block of rows we give the results of the default setting with IB1-IG and with a varying k parameter (number of nearest neighbours). We can see that a larger k improves performance to a certain extent.

In the second series of experiments we have used the MVDM metric. Here, we also varied the value of k. We found that a larger k yielded better results. In a variant on this series we applied MVDM only to the POS features. As expected this variant gave slightly better results.

In the third series we unpacked the features. Compared to the previous experiment the results were worse. Apparently, sparseness results in bad feature weights. This negative effect appears to have outweighted any positive effect of informative individual features.

In the last experiments we used Ripper to generate 390 complex features. The results are comparable to the best TiMBL settings.

In Table 2 we give an overview of the precision, recall and $F_\beta = 1$ of one of the best scoring setting: IB1-IG with $k = 3$

test data	precision	recall	$F_{\beta=1}$
ADJP	68.73%	63.24%	65.87
ADVP	79.85%	75.06%	77.38
CONJP	19.05%	44.44%	26.67
INTJ	100.00%	50.00%	66.67
LST	0.00%	0.00%	0.00
NP	90.69%	92.86%	91.76
PP	96.00%	97.19%	96.59
PRT	80.22%	68.87%	74.11
SBAR	85.77%	77.76%	81.57
VP	91.86%	92.74%	92.30
all	91.05%	92.03%	91.54

Table 2: Overview of the precision, recall and $F_\beta = 1$ of IB1-IG with $k = 3$ and MVDM.

4 Discussion

We found in the experiments that minor improvements on the default settings of TiMBL can be obtained by applying MVDM, particularly to the POS tags. A larger k generally improved accuracy to a certain extent. Unpacking the features did not give the expected improvement. Complex features, however, did, and seem a promising alley to go.

References

D. W. Aha, D. Kibler, and M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

S. Buchholz, W. Daelemans, and J. Veenstra. 1999. Cascaded grammatical relation assignment. In *Proceedings of EMNLP/VLC-99*, pages 239–246, University of Maryland, USA.

P. Clark and T. Niblett. 1989. The CN2 rule induction algorithm. *Machine Learning*, 3:261–284.

W. W. Cohen. 1995. Fast effective rule induction. In *Proc. of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California.

S. Cost and S. Salzberg. 1993. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine Learning*, 10:57–78.

W. Daelemans, A. Van den Bosch, and A. Weijters. 1997. IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2000. TiMBL: Tilburg Memory Based Learner, version 3.0, reference manual. Tech. Rep. ILK-0001, ILK, Tilburg University.

M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

J.R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning*, 1:81–206.

J.R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

C. Stanfill and D. Waltz. 1986. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, December.

A. Van den Bosch and J Zavrel. 2000. Unpacking multi-valued features and classes in memory-based language learning. In *Proceedings of ICML2000*, Stanford University, CA, USA.