Improving sequence segmentation learning by predicting trigrams

Antal van den Bosch
ILK / Computational Linguistics and AI
Tilburg University
Tilburg, The Netherlands
Antal.vdnBosch@uvt.nl

Walter Daelemans
CNTS, Department of Linguistics
University of Antwerp
Antwerp, Belgium
Walter.Daelemans@ua.ac.be

1 Introduction

Symbolic machine-learning classifiers are known to suffer from near-sightedness when performing sequence segmentation (chunking) tasks in natural language processing: without special architectural additions they are oblivious of the decisions they made earlier during the sequence, when making new ones. This near-sightedness might easily cause the classifier to produce invalid or impossible output sequences. This well-known problem has triggered at least the following three main types of solutions. (1) **Feedback**: Each training or test example may represent not only the regular windowed input, but also a copy of previously made classifications, to allow the classifier to be more consistent with its previous decisions. (2) **Stacking**: The partly incorrect concatenated output sequence of a single classifier may serve as input to a second-stage classifier in a stacking architecture, that learns to correct errors of the first-stage classifier. (3) **Output sequence optimization**: Classifier output may be augmented by an optimization over the output sequence as a whole using optimization techniques such as beam search or Viterbi search. In this paper we introduce a pointwise symbolic machine-learning classifier that predicts series of overlapping trigrams of class symbols, and have a simple voting mechanism decide on the final output sequence based on the overlapping predicted trigrams.

2 Data and methodology

The three data sets we used for this study represent a varied set of sentence-level natural language chunking tasks of both syntactic and semantic nature: English base phrase chunking (henceforth CHUNK), English named-entity recognition (NER), and disfluency chunking in transcribed spoken Dutch utterances (DISFL). All three tasks are to identify chunks, i.e., multi-word units of words carrying a particular label. In the CHUNK task, the label identifies a label of syntactic constituents (e.g., noun phrase, verb phrase, etc.). In the NER task, chunks are proper names of persons, locations, organizations, and miscellany other names. In the DISFL task, chunks are parts of spoken utterances that do not belong to the core syntactically correct utterance, such as hesitations, unfinished words, filled pauses, and stutters. We perform our experiments on the three tasks using three machine-learning algorithms: the memory-based learning or k-nearest neighbor algorithm, maximum-entropy classification, and Winnow learning in a sparse network.

3 Predicting class trigrams

Figure 1 exemplifies how sequences of input symbols, *windows*, are mapped to trigrams of class symbols. Consecutive positions in the input sequence produce overlapping trigrams. A classifier that learns to produce trigrams of class labels will at least produce syntactically valid trigrams from the training material, which might partly solve some near-sightedness problems of the single-class classifier. Although simple and appealing, the lurking disadvantage of the trigram idea is that the number of class labels increases explosively when moving from single class labels to wider trigrams.

To resolve the overlapping sequence of trigrams into the desired sequence of oiutput symbols, we vote over them using the following procedure: (1) When all three votes are unanimous, their common class label is returned; (2) When two out of three votes are for the same class label, this class label is returned; (3) When all three votes disagree (i.e.,

	MBL			MAXENT			WINNOW		
Task	Baseline	Trigram	red.	Baseline	Trigram	red.	Baseline	Trigram	red.
CHUNK	91.9	**92.7**	10	90.3	**91.9**	17	**89.5**	88.3	-11
NER	77.2	**80.2**	17	47.5	**74.5**	51	68.9	**70.1**	4
DISFL	77.9	**81.7**	17	75.3	**80.7**	22	**70.5**	65.3	-17

Table 1: Comparison of generalization performances of three machine-learning algorithms in terms of F-score on the three test sets without and with class trigrams.

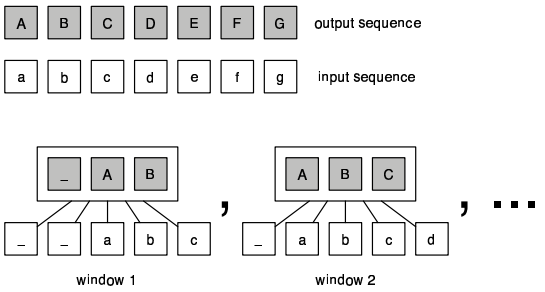

Figure 1: Windowing process with class trigrams.

Task	Stacking	Trigram	Combination
CHUNK	92.6	92.8	**93.1**
NER	78.9	80.2	**80.6**
DISFL	81.6	81.7	**81.9**

Table 2: Comparison of generalization performances in terms of F-score by MBL on the three test sets, with stacking, trigram classes, and the combination of the two.

when majority voting ties), the class label is returned of which the classifier is most confident. The results of the experiment in which we apply a near-sighted baseline classifier versus the trigram classifier to the three tasks, are listed in Table 1. We find rather positive effects of the trigram method both with MBL and MAXENT; we observe relative error reductions in the F-score on chunking ranging between 10% and a remarkable 51% error reduction, with MAXENT on the NER task. The performance of WINNOW is hurt by class trigrams, appararently due to WINNOW not being able to handle many sparse classes.

4 Trigrams versus feedback and stacking

We compare the trigram method to a feedback method and to a stacking method, using MBL as the base classifier. We find that similar error reductions are obtained with all three methods. We then combine the trigram method with feedback and stacking. The combination of the trigram method and feedback leads to increases in error. We also combine stacking with the trigram method. Table 2 compares the results of stacking and trigram classes with those of the combination of the two. The combination produces better results than the two methods individually, on all three tasks. Compared to the baseline single-class classifier, the error reductions are 15% for CHUNK, 15% for NER, and 18% for DISFL.

5 Conclusion

To combat near-sightedness in point-wise classifiers, we have proposed a new method that forces a point-wise classifier to predict trigrams of classes, using a simple voting mechanism to resolve the sequence of predicted overlapping trigrams to a sequence of single output symbols. This method can be applied to any point-wise classifier; we tested MBL, MAXENT, and WINNOW. Compared to their near-sighted baseline counterparts, error reductions are attained of 10 to 51% with MBL and MAXENT on three chunking tasks. We found weaker results with a WINNOW classifier, suggesting that the latter is more sensitive to the division of the class space in more classes. We observed that the positive effects of the trigram-class and stacking variants do not mute each other when combined, yielding the best error reductions of 15% to 18%.

Reference

This paper will appear in I. Dagan and D. Gildea (Eds.), *Proceedings of the Ninth Conference on Computational Language Learning*, CoNLL-2005, June 29–30, 2005, Ann Arbor, MI.