Memory-Based Named Entity Recognition

Erik F. Tjong Kim Sang
CNTS - Language Technology Group
University of Antwerp
erikt@uia.ua.ac.be

1 Introduction

We apply a memory-based learner to the CoNLL-2002 shared task: language-independent named entity recognition. We use three additional techniques for improving the base performance of the learner: cascading, feature selection and system combination. The overall system is trained with two types of features: words and substrings of words which are relevant for this particular task. It is tested on the two language pairs that were available for this shared task: Spanish and Dutch.

2 Approach

In this section we will give a brief description of the basic techniques employed in our approach to named entity recognition. We will describe memory-based learning, cascading, feature selection and system combination.

We use a nearest neighbor memory-based learner as a basic classifier (Daelemans et al., 2001). The learner stores all training data and classifies new data items by comparing them with the training data. The new data item will receive the same classification as that of the training item that is most similar to it. The data items are represented with symbolic features for which the learner computes weights which are based on their relevance for the classification task. The memory-based software package we used is called Timbl (Daelemans et al., 2001). We use its default learning algorithm, instance-based learning with information gain weighting (IB1IG), with the default setting of parameters (for example k=1).

The task of the learner is to predict the positions of the named entities in a text. The entities have been encoded with so-called IOB tags. These are tags which show that a word is outside of any entity (O), inside an entity (I) or at the beginning of an entity (B). For example, the sentence *John Smith called .* has the associated tags B-PER I-PER O O. This means that *John* starts a named entity of type PER, *Smith* continues this entity and neither *called* nor the final period are part of a named entity. The task of a system processing this data is to predict the sequences of entity tags as well as possible.

The initial set of parameters which we use for the learner for predicting the best entity tag for a word consists of the word and a group of preceding and following words. For example, we could use *John*, *Smith* and *called* as features when trying to find the best tag for *Smith* in the example sentence. In that case we would be using a left context of one word (in this case *John*) and a right context of one word (here *called*). It would be useful to know if the context words could be part of a named entity as well. In order to obtain this information, we perform CASCADING, feeding the output of one learner to the input of another learner (Veenstra, 1998). We will train a classifier on this task by using basic features such as words and then use the output tags of this system as input features for a second learner. For practical reasons we will only use the class tags of the context words and not that of the focus word. For example, the second system could represent the word *Smith* in the example sentence with five features: John, Smith, called, B-PER and O.

In general using features from a large context will give more detailed information about a word. However, the performance of memory-based learners can suffer when they need to process data with many features (Tjong Kim Sang, 2002). Since we do not know what word context features are best for this task, we will attempt to find the best set automatically by performing FEATURE SELECTION. There are too many different feature sets to perform a complete search. We will use a search method called bi-directional hill-climbing (Caruana and Freitag, 1994) for exploring the feature space. This method starts from a set of features (in our case the empty set) and compares the performance of a learner using this set with learners using the set with an extra feature or with one feature less. When the algorithm finds a feature set that enables the learner to perform better then it performs another search with this feature set. This procedure is repeated until the performance of the current feature set cannot be improved.

In our work on the CoNLL-2000 shared task of chunking (Tjong Kim Sang, 2002), we have shown that performance on a phrase classification task can be improved by performing SYSTEM COMBINATION. Since named entity classification is similar to chunking, we will use this technique here as well. We will use the same approach as described in Tjong Kim Sang (2002): apply one learning technique to five different representations of the output tags: IOB1, IOB2, IOE1, IOE2 and O+C. Changing the output representation will change the task of the learner. For different output representations it will make different errors. We will convert the output for the different data representations to one data representation (O+C) and for each word select the tag that has been predicted most often (majority voting).

In (Tjong Kim Sang, 2002), we have evaluated three different processing strategies for finding chunks in text: 1. predicting chunk boundaries and chunk classes simultaneously, 2. predicting boundaries first and classes later, and 3. building a separate recognizer for each different class. We chose the second processing strategy because it required fewer computer resources than the third and performed better than the first. We use this approach here as well: first we attempt to find the boundaries of named entities and then we compute the most likely class for the entities that have been found.

3 Results

In our first approach to named entity recognition, we have applied the chunker described in Tjong Kim Sang (2002) to the Spanish data. This data set did not have part-of-speech tags available so we have only used words as features. Each word was represented by itself and the three preceding and the three next words. We determined the best parameters for our approach by performing experiments with the training data for Spanish while using 10-fold cross-validation. For this purpose the data was divided in ten parts of approximately the same size and each part was processed while using the other nine as training data.

We started with removing the entity class information from the data, keeping entity borders only. For each of the five available output representations we have performed a feature selection process for finding an optimal set of features for this task. Each of the results was fed to cascaded system which had access to the seven word features as well as the predicted class tags for the two words before the focus word and the two words following the focus word. The results of the five cascaded systems were converted to brackets (O+C representation) and these were combined with majority voting. We evaluated all combinations of three, four and five systems and choose the best. Finally, classes were added to the resulting entities by a learner which had access to the first and the last word of the entity plus three words before the first word and three words behind the last. Again, feature selection was used for finding the best feature set. The output of the learners was evaluated with $F_{\beta=1}$ rates which are based on the precision and recall of entities (van Rijsbergen, 1975).

In almost all cases the feature selection method used only a subset of the available features (seven word features and four additional tag features). The cascaded systems outperformed the base systems in three of the five cases. A majority vote of the result was always better than the best of the individual systems (at best $F_{\beta=1}$=79.40 for the cascaded systems). After determining the categories of the named entities, performance dropped to $F_{\beta=1}$=71.45.

A problem of our current approach is that the system has few clues for handling new words. In the CoNLL-2000 chunking task, the part-of-speech tags helped to classify these but in the Spanish data there is no additional information about the words available except from their context. It seems reasonable to use word-internal morphological information as a clue. This could

Spanish train	Pass 1			Pass 2			
Representation	$F_{\beta=1}$	features used		$F_{\beta=1}$	features used		
IOB1	85.86	$w_{-2..0}$	$m_{fp,fs,ps}$	88.68	$w_{-2,0,1}$	$t_{-1,1}$	$m_{fp,fs,ps}$
IOB2	82.14	$w_{-2..1}$	$m_{fs,pp,ps}$	84.39	$w_{-1..1}$	$t_{-2,-1,1}$	$m_{pp,ps}$
IOE1	85.86	$w_{-2..0}$	$m_{fp,fs,ps}$	88.76	$w_{-2,0,1}$	$t_{-1,1}$	$m_{fp,fs,ps}$
IOE2	77.18	$w_{-2..2}$	$m_{fp,fs,pp}$	83.50	$w_{-1,0,2}$	$t_{-1,1}$	$m_{fs,pp}$
O+C	80.33	O: $w_{-2..0}$	$m_{fp,pp,ps}$	84.08	O: $w_{-2..0}$	$t_{-2,-1}$	$m_{fp,pp,ps}$
		C: $w_{-2..1}$	m_{ps}		C: $w_{-1..2}$	$t_{1,2}$	m_{ps}
Majority voting	86.10	O: IOB1 IOB2 IOE1 IOE2 O		88.96	O: IOB1 IOB2 IOE1 IOE2 O		
		C: IOB1 IOB2 IOE1 IOE2 C			C: IOB1 IOB2 IOE1 IOE2 C		
with classes	72.29	$w_{-2,-1,\text{start,final}}$	$m_{start.s}$	74.34	$w_{-2,-1,\text{start,final}}$		$m_{start.s}$

Table 1: $F_{\beta=1}$ rates for identifying entity borders (not entity types) with word features (w) and morphological features (m, see text below) only from the Spanish training data, processing with 10-fold cross-validation while five different output data representations (IOB1, IOB2, IOE1, IOE2 and O+C), cascading with extra classification tag features (t), feature selection and system combination. The best results are obtained by using only a limited number of the available features ($w_{-3..3}$, $t_{-2,-1,1,2}$ and $m_{fp,fs,pp,ps}$). Cascading (pass 2) generally improves performance when compared with pass 1. Majority voting performs better than any of the individual learners while using only a few of their results. The bottom line shows the performance after adding class information.

be done by using the first few characters or the last few characters of a word as an extra feature. The problem is that we do not know how many characters we have to select to get useful features. Some character sequences of a specific length might be interesting for this task while others of the same length might be not. We have decided to use a statistical measure for selecting morphological strings that are useful for performing this task in a particular language.

The statistical measure which we have chosen selects all prefix and suffix strings that appear in the training data in capitalized words ten times or more and additionally are part of a word in a named entity in 95% of the cases or more. Examples of such strings in the Spanish data are *Europea* (prefix, appears 61 times and is a part of named entity words in 98% of the cases) and *z* (suffix, appears 734 times and 99% of the time in named entity words). The system looked for phrases in the Spanish training data of ten characters or shorter and found 790. The word immediately in front of a named entity word can be an important clue and therefore we have extracted similar prefix and suffix features for words immediately before capitalized named entity words. Examples from the Spanish training data are the prefix *una* (24 times, 100%) and the suffix *e* (5042 times, 100%). For this particular type, 214 strings were selected.

We have added four morphological features to the data: focus word prefix (fp), focus word suffix (fs), previous word prefix (pp) and previous word suffix (ps). After this we repeated the 10-fold cross-validation experiment with the Spanish training data. For each of the 10 parts, the morphological features were generated from the other nine parts only. The results of this experiment can be found in Table 1. Morphological features were chosen as useful features in all cases. The most frequently chosen feature was the suffix of the previous word (ps). The maximum performance of the unlabeled task when compared with using word features only, improved considerably, from 79.40 to 88.96 (46% error decrease). The increase after adding categories to the named entities was smaller: from 71.45 to 74.34 (10% error decrease).

We have used the best configuration found for the Spanish training data for processing both the Spanish test sets and a configuration obtained from the Dutch training data (without the part-of-speech tags) for the Dutch test sets. The results for processing the test data sets can be found in Table 2. Overall precision is always higher than overall recall but the difference is never larger than 4.4 percentage points. For both languages the system performs better on the test data than on the development data.

4 Concluding Remarks

We have presented a machine learning method for performing language-independent named entity recognition. It uses a memory-based classifier as base learner. The performance of this learner is improved with cascading, feature selection and system combination. The system uses both words and prefixes and suffixes of words. The latter two are derived with a statistical method which selects substrings of words which frequently appear in words in or near named entities. The learner had no access to linguistic information other than that was made available in the training data. Its performance was not as good as state-of-the-art named entity recognizers for English (over $F_{\beta=1}$=90, see for example Mikheev (1998)). However, it performs reasonable on the two languages in this shared task ($F_{\beta=1}$=75 for the Spanish test set and $F_{\beta=1}$=70 for Dutch).

The strength of our system is its ability to operate without many linguistic clues about the language that is processed. A text with annotated entities is enough to obtain a reasonable performance. A practical weakness is its processing speed: the current implementation processes only about 6 words per second on a parallel machine. Another weakness is the selection of morphological features. This relies on the fact that many named entity words in the two target languages are capitalized, a feature which may not help for other languages (for example German and Hindi). We believe that a further improvement of the performance of the system could be obtained by using features derived from interesting statistical information from the training text and perhaps even from other untagged text.

Spanish dev.	precision	recall	$F_{\beta=1}$
LOC	67.90%	81.83%	74.22
MISC	51.76%	42.92%	46.93
ORG	77.08%	70.24%	73.50
PER	86.90%	77.09%	81.70
overall	74.79%	71.99%	73.36

Spanish test	precision	recall	$F_{\beta=1}$
LOC	76.01%	76.01%	76.01
MISC	63.70%	50.59%	56.39
ORG	76.45%	78.36%	77.39
PER	79.57%	81.09%	80.32
overall	76.00%	75.55%	75.78

Dutch devel.	precision	recall	$F_{\beta=1}$
LOC	79.21%	72.06%	75.47
MISC	68.63%	65.68%	67.12
ORG	76.13%	49.78%	60.20
PER	62.61%	80.65%	70.49
overall	69.60%	66.77%	68.15

Dutch test	precision	recall	$F_{\beta=1}$
LOC	83.21%	73.28%	77.93
MISC	72.79%	64.45%	68.36
ORG	75.63%	54.50%	63.35
PER	65.72%	81.97%	72.95
overall	72.56%	68.88%	70.67

Table 2: Results obtained for the development and the test data sets for the two languages used in this shared task.

Acknowledgements

Tjong Kim Sang is supported by IWT STWW as a researcher in the ATRANOS project.

References

Rich Caruana and Dayne Freitag. 1994. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36. New Brunswick, NJ, USA, Morgan Kaufman.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2001. *TiMBL: Tilburg Memory Based Learner, version 4.0, Reference Guide.* ILK Technical Report ILK-01-04. http://ilk.kub.nl/.

Andrei Mikheev, Claire Grover, and Marc Moens. 1998. Description of the ltg system used for muc-7. In *Seventh Message Understanding Conference (MUC-7).* Fairfax, Virginia.

Erik F. Tjong Kim Sang. 2002. Memory-based shallow parsing. *Journal of Machine Learning Research*, 2(Mar):559–594.

C.J. van Rijsbergen. 1975. *Information Retrieval.* Buttersworth.

Jorn Veenstra. 1998. Fast np chunking using memory-based learning techniques. In *BENELEARN-98: Proceedings of the Eighth Belgian-Dutch Conference on Machine Learning.* ATO-DLO, Wageningen, report 352.