Named-Entity Recognition in Novel Domains with External Lexical Knowledge

Massimiliano Ciaramita
Inst. of Cognitive Science and Technology
Italian National Research Council
m.ciaramita@istc.cnr.it

Yasemin Altun
Toyota Technological
Institute at Chicago
altun@tti-c.org

Abstract

We investigate the adaptation of structured classifiers to new domains. In particular, the problem of using a supervised Named-Entity Recognition (NER) system on data from a different source than the training data. We present a Semi-Markov Model, trained with the perceptron algorithm, coupled with an external dictionary with the goal of improving generalization on the novel domain. Preliminary experiments show promising results, obtained with very simple additional features.

1 Introduction

Named-entity recognition (NER) is the task of tagging words with labels such as "person", "organization", and "location". In the standard supervised setting the task can be solved accurately with machine learning techniques, in particular sequence learning methods. NER has been successful on newswire, in different languages, and biomedical text (cf. [XMP02, FIJZ03, DFN$^+$05]). It can provide crucial, although shallow, semantic information for tasks ranging from question answering to anaphora resolution. Since manually annotated data is rarely available, it is natural to ask how accurate NER systems are, as an off-the-shelf technology, at tagging data different from the available training data.

Applying an NER system to a novel domain can yield a dramatic accuracy loss. This might be partially due to inconsistencies in the manual tagging procedure; e.g., in domains which require very specific expertise, such as molecular biology (cf. [DFN$^+$05] on this problem). Another problem is that names in different domains have different morpho-syntactic properties; i.e., they look different and occur in different contexts. As far as the morphological aspect is concerned we propose an approach based on coupling an off-the-shelf supervised NER system with an external dictionary. The model is a Semi-Markov Model, introduced in [CS04], which provides a suitable framework for including external knowledge in the classifier, which we compare with a perceptron-trained HMM (P-HMM).

We train our models on the CoNLL 2003 dataset and evaluate them on a manually annotated section (Section 00) from the Wall Street Journal portion of the Penn Treebank [MSM93]. We show how the performance on the novel data is improved by coupling the system with a domain-independent dictionary, and simple string similarity features. In Section 2 we illustrate the problem of the performance degradation. In Section 3 we describe the Semi Markov Model (P-SMM) and the dictionary features. In Section 4 we discuss our results.

2 Performance Degradation of an NER system

Let us assume the existence of a supervised classifier trained for an NER task. One might want to use such system as an off-the-shelf tool on a new domain (text), even though the text might come from a difference source than the original training data – as long as the task (i.e. the label set) is the same. For example, the goal could be to find people, organizations and locations names in the Wall Street Journal with a tagger trained on the manually annotated portion of the Reuters newswire corpus. Unfortunately, it turns out that even for such – relatively – similar types of texts the performance of a supervised classifier degrades significantly. To quantify this effect we implemented an HMM model, trained with the perceptron algorithm (inspired by that of [Col02]). The P-HMM is trained on the CoNLL 2003 English dataset. The model uses standard contextual and morphology-spelling features:

- word features: w_i, w_{i-1}, w_{i+1}, $w_i + w_{i-1}$, $w_i + w_{i+1}$
- part of speech: pos_i, pos_{i-1}, pos_{i+1}, $pos_i + pos_{i-1}$, $pos_i + pos_{i+1}$
- substrings: prefixes and suffixes up to 6 characters of w_i
- word shape(1 and 2): s_i, s_{i-1}, s_{i+1}, $s_i + s_{i-1}$, $s_i + s_{i+1}$

Shape-1 is a regular expression-like transformation in which each character c of a string is substituted with X if c is capitalized, with x if c is lowercase, with d if c is a digit and with c itself otherwise. Shape-2 is a transformation of shape-1 in which each sequence of two or more characters c is substituted with $c*$. For example, if s = "Merrill Lynch& Co.", shape-1(s) = "Xxxxxxx Xxxxx & Xx.", and shape-2(shape-1(s)) = "Xx* Xx* & Xx*.".

Evaluation on the CoNLL dataset was conducted by 5-fold cross-validation. The data is split in three partitions: training (50%), test (33%) and development (17%). We used the label set {0,PER,ORG,LOC,MISC}. Each label "X", other than "0", is split into "B-X" (beginning) and "I-X" (continuation). The development set is used to fix the number of times the training data is processed, T. The model achieves an F-score of 0.908 (0.0034% standard error). Similarly we split the manually annotated Section-00 of the Penn Treebank, henceforth WSJ-00, in test (66%) and development (34%), trained the model on the full CoNLL data, and used the development partition of WSJ-00 to fix T. In this second experiment F-score drops to 0.643 (0.0095 standard error).

3 Semi Markov Models with Dictionary Features

In [SC04, CS04], it has been shown that Semi-Markov methods are a natural way of exploiting dictionaries in NER tasks. In particular, SMMs enable features that encode similarities between two sequence segments of arbitrary lengths. In NER tasks, these features correspond to similarity measures between a word sequence to be classified and a name coming from a pre-compiled list or an available lexical resource. This type of model is particularly useful in the face of sparse data. The question we investigate here is whether this model is also beneficial to improve tagging on novel datasets. Here the system is provided with a list of entity names extracted from the sense-annotated portion of the Brown corpus [MLTB93]. All strings whose part of speech is "NNP" or "NNPS" and word sense tag is either "person", "group", "location" or "other" are included in a dictionary called SEMCOR. Each entry is also associated with its shape-1 and shape-2 forms.

3.1 Perceptron Training for Semi Markov Models

We are interested in learning a discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over observation/label sequence pairs where F is linear in a feature representation Φ defined over the

joint input/output space

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle. \quad (1)$$

Given a new observation sequence $\mathbf{x}$, we make a prediction by maximizing this function over the response variable

$$f(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}). \quad (2)$$

Following [SC04], we define the input space as $\mathcal{Y} \subset (Z^* \times Z^* \times Y)^+$ and a label sequence $\mathbf{y} \in \mathcal{Y}$ as a sequence of segment labellings $s_i = (b_i, e_i, y_i)$ where b_i and e_i denote the beginning and the end of the segment whose label is given by $y_i \in Y$, where Y is the set of individual labels (e.g. Person, Location) and Z^* is the set of non-negative integers.

$$\mathcal{Y} = \{\mathbf{y} = (s_0, \ldots, s_n) | i = \{0, \ldots, n\},\ b_i \leq e_i\ \wedge\ b_i = e_{i-1} + 1\} \quad (3)$$

with the convention that $e_{-1} = -1$.

In SMMs, Φ extracts three kinds of features from the observation/label sequence pairs: features that encode interactions between attributes of the observation sequence and the label of a *segment* (rather than the label of an observation as in HMM); features that encode interactions between neighboring labels along the sequence; features that encode properties of a segment. The first two types of features are commonly used in other sequence models, such as HMMs and Conditional Random Fields (CRFs). The third feature type, explained in more details below, is specific to Semi-Markov models.

We perform average-perceptron training (Algorithm 1), which is a simple extension of the algorithm given in [CS04]. $\hat{\mathbf{y}}$ can be found by the Viterbi algorithm which searches over all possible segment assignments. Its complexity is $O(nl)$ where n is the length of the observation sequence and l is the maximum length of a segment.

Algorithm 1 Semi Markov Average Perceptron algorithm.

1: Initialize $\mathbf{w}_0 = \vec{0}$.
2: **for** $t = 1, \ldots, T$ **do**
3: Select $\mathbf{x}^i$ and compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}^i, \mathbf{y}; \mathbf{w})$
4: **if** $\mathbf{y}^i \neq \hat{\mathbf{y}}$, $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Phi(\mathbf{x}^i, \mathbf{y}^i) - \Phi(\mathbf{x}^i, \hat{\mathbf{y}})$
5: **end for**
6: **return** $\mathbf{w} = \frac{1}{T} \sum_t \mathbf{w}_t$

3.2 Dictionary features

Similarly to [CS04], in addition to the features described in Section 1, the SMM uses features which represent properties of whole segments rather than single words, such as the similarity between the segment and dictionary entries for the same label. More precisely, for each segment s and label y we include as a feature the distance, approximated to the second decimal digit, of the most similar entry in the dictionary for label y. We compute the minimum distance not only for words $s = w_u + .. + w_v$, u and v being the start and ending of s, but also for the more general forms $s = s1(w_u) + .. + s1(w_v$ and $s = s2(s1(w_u)) + .. + s2(s1(w_v))$.

As a string similarity measure we use the Jaccard distance: given two sets S and T $jaccard(S, T) = |S \cap T|/|S \cup T|$[1]. We consider both the sets of characters and sets of words in the segment. As an example, the segment "George Duffield", at the character

[1]Other string similarity measures such as Jaro-Winkler and edit distance, or measures of distributional association can be easily encoded in this model.

Model	Train	Test	Dictionary	F-score	Std. error
P-HMM	CoNLL	CoNLL	-	0.908	0.0034
P-HMM	CoNLL	WSJ-00	-	0.643	0.0095
P-SMM	CoNLL	CoNLL	SEMCOR	0.906	0.0067
P-SMM	CoNLL	WSJ-00	SEMCOR	0.691	0.0096

Table 1. Summary of results for the Perceptron HMM and SMM models with mean F-score and standard error computed with 5-fold cross-validation.

level, has features such as $min^{PER}_{char-w} = 0.69$ (since "George Dillon" is in the dictionary) and $min^{ORG}_{char-w} = 0.65$ ("Florida Grapefruit League"), while at the word level it has features such as $min^{PER}_{word-w} = 0.5$ ("George") and $min^{PER}_{word-s2} = 1$ ("Xx* Xx*"). We also include segment length features; e.g. $l(\text{"}George\ Dufield'') = 2$.

4 Results and Conclusion

Table 1 summarizes the results of our experiments. In the cross-validation setting the P-HMM and P-SMM models produce comparable results. When tested on the WSJ-00 their accuracy drops dramatically. However the model supported by the SEMCOR dictionary is more accurate then the unsupported P-HMM in terms of F-score by almost 5%. This improvement does not guarantee enough accuracy for the practical purpose of applying NER to novel domains. However, it suggests an interesting line of research in which taggers are not parametrized only with respect to the available training data, which is inevitably biased, but also with respect to an external "ontology" which intuitively acts as a bridge towards data from different sources. As future research we plan to evaluate the impact of different dictionaries, which might be tailored to the characteristics of the data to be tagged; e.g., in this case a dictionary of entity names in the financial-business domain.

References

[Col02] M. Collins. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Empirical Methods of Natural Language Processing (EMNLP)*, 2002.

[CS04] W. Cohen and S. Sarawagi. Exploiting dictionaries in named entity extraction: Combining semi-markov extraction processes and data integration methods. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2004.

[DFN+05] S. Dingare, J. Finkel, M. Nissim, C. Manning, and C. Grover. A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. *Comparative and Functional Genomics*, 6:77–85, 2005.

[FIJZ03] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named Entity Recognition through Classifier Combination. In *Proceedings of the Conference on Natual Language Learning (CoNLL 2003)*, 2003.

[MLTB93] G.A. Miller, C. Leacock, R. Tengi, and R.T. Bunker. A Semantic Concordance. In *Proceedings of the ARPA Workshop on Human Language Technology*, 1993.

[MSM93] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 1993.

[SC04] S. Sarawagi and W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[XMP02] Carrerars X, L. Marques, and L. Padro. Named Entity Extraction Using Adaboost. In *Proceedings of the Conference on Natual Language Learning (CoNLL 2002)*, 2002.