Named entity recognition without domain-knowledge using conditional random fields

Felix Jungermann FELIX.JUNGERMANN@UDO.EDU
University of Dortmund, Baroper Strasse 301, Campus Sued (GB IV), 44227 Dortmund, Germany

Abstract

This paper addresses the problem of not using any domain-knowledge in named entity recognition (NER) tasks. Experiments on two well-known datasets show that the currently mostly used technique – conditional random fields (CRF) – achieves results which are respectable. It is discussed if it is acceptable to pass on better results to get results in a faster and modular way.

1. Conditional Random Fields

CRFs are undirected graphical models (Lafferty et al., 2001). In theory of CRFs a set of states Y is globally conditioned by an observation sequence X. In the case of NER the states are the alphabet used for tagging text sequences. The text sequences on the other hand are the so called observation sequences. The probability of a state sequence being the best for a given observation sequence is calculated by potential functions. There are two different kinds of potential functions: state-features and transition-features. State-features are used to examine states and corresponding observations at one position of the sequence. An exemplary state-feature is: "The observation Hamburg leads to the state 'Location'" Transition-features are used to analyse transitions between states and the observations that lead to such transitions. An exemplary transition-feature is: "The transition from null to Person is done if the first word is 'Mr.' and the second word is any other word".

2. Implementation for Yale

The implementation to be described is based on an existing java-implementation of CRFs (Sarawagi & Cohen, 2004) and extends it for universality. Finally the implementation was integrated into the learning environment Yale (Mierswa et al., 2006) as a plugin for textmining. In many textmining-environments the "bag-of-words"-representation is the mostly used technique to examine texts (for classification for instance). That situation also was existent in Yale. But for NER it is necessary to conserve the sequence-character of texts – in contrast to a "bag of word"-solution which ignores such characteristics. Therefore the author designed an environment to access and augment texts in Yale. In addition to the possibility to store texts in a sequential way, some basic features of texts were defined to extract them out of the text for using them as features in the CRF. Some of these features are for example ngrams, prefixes, suffixes and so on. The expectation is that such features can be used for every kind of text so no explicit domain-knowledge is needed. A user has the possibility to handle text-analysis-tasks in a modular way: load texts into Yale, enrich them by additional features like all ngrams of the text and finally use a learner for texts – in this case CRF. Such modularity can easily be extended by new feature-extraction-operators for example. But the implementation – in its current state – is just a first step. The goal is to develop a more modular text-analysis-environment which can – by using machine learning techniques – determine the best selection of features by its own.

3. NER without domain-knowledge

In past work (Jungermann, 2006) the NER-tasks of the JNLPBA[1]2004 and of the CoNLL[2]2003 were analyzed. Different features like n-grams, prefixes, suffixes, regular expressions and so on were used. The results are shown in Table 1. In addition to the results of the author the best results – achieved by the conference-participants – are shown. The 'position'-column ranks the results of the author among the number of results of the conference-participants.

[1] Joint workshop on natural language processing in biomedicine and its applications

[2] Conference on computational natural language learning

[3] The results are given in % f-measure.

Table 1. Results on NER-tasks

Data set	Jungermann [3]	best [3]	position
JNLPBA 04	64.9	72.6	5/9
CoNLL 03	60.7	72.4	14/17

3.1. Used domain-knowledge

The achieved results are respectable but not good. If one analyses the systems that result in better performance one can see that better performance often is an outcome of using domain-knowledge. It was fundamental for the CoNLL03 to use external and additional non-tagged information(-sources) to augment the NER-task. At the JNLPBA-task most of the participants benefit from external information(-sources). (Settles, 2004) for example used up to 17 dictionaries which were created (seven of them manually) using domain-knowledge. These dictionaries for example contain greek letters, which indicate a protein, amino acid or chemical elements and so on. Settles also used the search-engine Google to get auxiliary and up-todate information out of the internet. The actual system described by this paper does not use any comparable information but just utilizes the training-data for building up dictionaries and feature-sets. If you keep that in mind, it is not surprising that the results are average.

4. Gaining domain-knowledge automatically

Domain-knowledge is needed to achieve got results. But a great amount of work is to be done to collect good domain-knowledge – even more if you are not a linguist. So it would be nice if one could avoid that working-step or if one could use an automatic way to gain such knowledge in order not to learn rules from the domain "manually". Using additional data like (Roessler & Morik, 2005) did is a similar approach which unfortunately still needs domain-knowledge to identify the texts that can be used to enrich the given data. Recent works on NLP like (Gabrilovich & Markovitch, 2007) and (Gabrilovich & Markovitch, 2006) for example use the online dictionary wikipedia for text categorization which has different advantages. First of all it is open source and free to use. Second, it is an up-to-date information-source because a lot of people are daily updating that dictionary. Finally the category system of wikipedia could be used as a kind of meta-information-source.

Future work of the author will point in a similar direction by using wikipedia (Cramer et al., 2007) to build up gazetteers to augment NER-tasks.

References

Cramer, I., Jungermann, F., Mehler, A., & Gleim, R. (2007). Extracting an application-oriented wikipedia corpus. *To be submitted at WAC3.*

Gabrilovich, E., & Markovitch, S. (2006). Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. *Proceedings of the Twenty-First National Conference on Artificial Intelligence.*

Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. *IJCAI* (pp. 1606–1611).

Jungermann, F. (2006). Named entity recognition mit conditional random fields. *Master thesis. University of Dortmund.*

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning.*

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006).*

Roessler, M., & Morik, K. (2005). Using unlabeled texts for named-entity recognition. *Proceedings of the ICML 2005 Workshop on Learning with Multiple Views.*

Sarawagi, S., & Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. *NIPS.*

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. *International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004* (pp. 107–110).