Modular Classifier Ensemble Architecture for Named Entity Recognition on Low Resource Systems

Christian Hänig **Stefan Bordag** **Stefan Thomas**
ExB Research & Development GmbH
Seeburgstr. 100
04103 Leipzig, Germany
[haenig|bordag|thomas]@exb.de

Abstract

This paper presents the best performing Named Entity Recognition system in the GermEval 2014 Shared Task. Our approach combines semi-automatically created lexical resources with an ensemble of binary classifiers which extract the most likely tag sequence. Out-of-vocabulary words are tackled with semantic generalization extracted from a large corpus and an ensemble of part-of-speech taggers, one of which is unsupervised. Unknown candidate sequences are resolved using a look-up with the Wikipedia API.

1 Introduction

Recognizing named entities in unstructured text in multiple languages and across different domains remains a challenging task. This can be gauged by the fact that for German the best Named Entity Recognition (NER) systems only achieve around 80% F1 (Faruqui and Padó, 2010). NER is even more difficult when resource limitations such as RAM usage or CPU time need to be taken into account, because then popular strategies such as simply using all possible character n-grams as features become infeasible. This is of particular importance when developing linguistic solutions for mobile platforms.

The relevant topics to cover when designing a NER system are which training data to use, which classifier to use and which features the classifier should be based on.

We present a NER system designed to minimize the impact of limited computational resources on the quality of the results and to maximize the cross-linguistic and cross-domain performance. This is implemented through a modular approach with complementary supervised components and unsupervised fall-back equivalents, ensuring adequate results even without part-of-speech (POS) annotated data.

2 Architecture of our solution

Our system consists of an ensemble of classifiers (see Section 2.1), list- (see Section 2.2) and pattern-based (see Section 2.3) annotators, and modules for the special treatment of out-of-vocabulary (OOV) words (see Section 2.4). Each module provides confidence scores for all annotations, which enables the ensemble to combine all candidate annotations to produce the most likely tag sequence (see Section 3).

2.1 Classifier-based annotation

Features typically encode aspects of either the target word or the surrounding words such as capitalization, part-of-speech or semantic information. In some languages, such as English, there are features which strongly indicate that the target word is a name, such as capitalization. Therefore NER systems for English typically achieve very good F1 scores of around 90% (e.g. 88.76% as reported by Sang and Meulder (2003)). In German, capitalization is used for all nouns and there are no such obvious features as strongly indicative as English capitalization (Tkachenko and Simanovsky, 2012).

2.1.1 Features

We extract the following features for each of the tokens, usually in a 5-word-window around the target token:

Words Plain token strings

POS tags Tags obtained by a supervised tagger (Stanford Tagger as described by Toutanova et al. (2003)) and tags obtained by an unsupervised tagger based on *SVD2* as described by Lamar et al. (2010)

Word Shape Shape features based on Bikel et al. (1999) and shape features that are used by the Stanford NER (Finkel et al., 2005)

Semantic Classes We compute semantically similar words and cluster them as described by Gamallo and Bordag (2011), and use the resulting classes as features.

Additionally, we extract all n-grams of the target word (Finkel et al., 2005) and for compound words, use their components (e.g. *Berlin/Deutschland* leads to two additional word features: *Berlin* and *Deutschland*).

2.1.2 Classifier selection

Typically, classifier NER systems use either Conditional Random Fields (CRF) (Finkel et al., 2005), Maximum Entropy classifiers (ME) (Borthwick, 1999) or other machine learning methods. Apart from differing slightly in their generalization power, the classifiers also differ in training time, classification time and RAM usage. One interesting question is, how a probably slightly better classification method such as CRF compares to MaxEnt regarding runtime and memory consumption. One of the relevant differences is that ME classifiers tag each token individually, CRFs (and other sequence models like HMMs (Leek, 1997) and CMMs (Borthwick, 1999)) use adjacent words as well (Lafferty et al., 2001).

We experimented with three different classifiers: A collection of binary CRFs, a collection of binary MEs and a collection of improved binary MEs with an additional name boundary classification method. We trained them on the training data of GermEval 2014 (Benikova et al., 2014) with the features described in Section 2.1.1 and evaluated against the GermEval 2014 development data. Each of the classifiers was trained for each of the three NER categories *LOC*, *ORG* and *PER*. We additionally extended the ME classifier with a Boundary Detection algorithm (ME-BD) to overcome its weaknesses in sequence tagging. Therefore, we trained two ME classifiers: one for the left boundary and one for the right boundary, respectively. Each extracted entity is then extended employing both boundary classifiers until the most likely boundary has been detected.

Table 1 summarizes our results:

Classifier	Class	P	R	F
ME	LOC	0.854	0.569	0.683
ME	ORG	0.559	0.438	0.491
ME	PER	0.701	0.488	0.576
ME-BD	LOC	0.867	0.581	0.695
ME-BD	ORG	0.696	0.516	0.593
ME-BD	PER	0.893	0.609	0.724
CRF	LOC	0.856	0.632	0.727
CRF	ORG	0.793	0.502	0.615
CRF	PER	0.849	0.743	0.792

Table 1: M1 Scores for different classifiers / categories

Boundary detection significantly improves the performance of ME classifiers, especially for categories whose entities often consist of multiple tokens (e.g. *ORG* and *PER*). It took 8 hours to train the CRFs compared to 1 hour for the ME classifiers. Although CRFs provide clearly superior results in this experiment, it is obviously not feasible to train CRF models on mobile devices.

2.2 List-based annotation

We created entity lists for three NER categories and a catch-all *OTH* for unclassified NEs, as well as a number of subcategories for each (see Table 2 for a selection of these categories).

After crawling multiple freely available sources (e.g. OpenStreetMap[1] and Wikipedia[2]), we manually revised all extracted items. The main objective of this step is to reduce ambiguity to retain only high confidence items.

[1] http://www.openstreetmap.org/

[2] http://www.wikipedia.org/

The resulting lists are augmented with inflections, synonyms and abbreviations. We extracted all candidate items from a large word list computed for a crawled web corpus that are semantically or orthographically similar to the seed item. Finally, the suggested candidate items were manually revised and added to the entity lists.

NER category	subcategories
LOC	astronomical locations, castles, cities, continents, countries, highways, islands, lakes, mountains, (historical) regions, rivers, schools, seas, states, streets
PER	artists, historical persons, politicians, scientists, sportspersons, VIPs
ORG	aircraft / automobile / phone manufacturers, sports associations, cellphone providers, companies, financial institutions, musical bands, newspapers, organizations / associations, parties, politically motivated groups, radio channels, sports teams, television channels, universities / research institutes
OTH	airplane / automobile / cellphone models, currencies, historical events, products

Table 2: NER categories and selected subcategories

The list-based matching process shows a preference for longer matches over short ones (e.g. *FC Bayern München* supersedes *Bayern München*) and assigns a confidence score to each annotation. Confidence scores are estimated for each category separately based on an evaluation against our internal data sets.

2.3 Pattern-based annotation

Our pattern framework allows creation of almost arbitrary patterns, for example:

Suffix patterns If a word is uppercase and ends with *stadt* or *hausen* or *ingen* then annotate it as *LOC*.

Complex patterns If a word contains a dot followed by a top level domain and ends after the domain or is followed by a punctuation character then annotate it as *URL*[3] .

Sequence patterns If an uppercase word is followed by *AG* or *GmbH* or *Inc.* then annotate both words as *ORG*.

All patterns may be combined with specific exclusions to prevent incorrect high frequency words from being annotated (e.g. *Hauptstadt*[4]). Another heuristic that is used for lexicon matching also holds for pattern matching: long sequence matches supersede short matches.

2.4 Classification of Out-Of-Vocabulary words

We employ several strategies to cope with out-of-vocabulary words.

This includes the computation of both semantic generalizations (Faruqui and Padó, 2010) and syntactic generalizations of the words in the target data set (see Section 2.1.1) based on a large German web corpus (produced by our web crawler, consists of about 50M sentences).

We also compute a list of valid string transformations between categories. For each pair of words, a string transformation is computed (e.g. *Italien* to *italienische* is *lower-case(0) + -ische*). All obtained transformations are ranked according to their frequencies, pruned and manually revised. During classification these rules are applied to unknown words to transform them into possibly known words. This was applied on the source category *LOC* and the target categories *LOC*, *LOCderiv* and *LOCpart*.

Finally, we extract sequences of entity candidates (e.g. out-of-vocabulary uppercase words) and use the Wikipedia API to get more information about the candidates if category information is available in Wikipedia.

3 Classifier ensemble

The annotators finally vote on the joint output of the ensemble by sorting all the annotations of a sentence in descending order according to their

[3] URLs are mapped to OTH for this task.

[4] Means *capital city* and is a common noun.

confidence scores. Shorter annotations are discarded in case of overlaps.

The combiner then iterates over the ranked annotations and adds the annotation with the highest score as outer entity to the final tag sequence. If it overlaps with a higher ranked annotation of another type then it is added as inner entity instead. Any other types of overlaps are discarded. These steps are repeated until each of the annotations either has been added to the final tag sequence or has been discarded by the combination method.

3.1 Evaluation results

We created three models: a CRF model with unlimited resources (CRF; model size: 271MB), a low-resource CRF model (mCRF; model size: 41MB without technical compression) and a ME-BD model (ME-BD; model size: 159MB). The feature space of the low-resource CRF model was pruned significantly by removing n-grams and Stanford POS tags completely. Furthermore, the tremendous amount of token features is reduced to the 10k most frequent German words.

We trained all three models on the joint set of training and development data. The official evaluation scores obtained by evaluation against the test set are provided in Table 3:

Model	Metric	P	R	F
CRF	M1	0.781	0.748	0.764
CRF	M2	0.789	0.755	0.771
CRF	M3 outer	0.807	0.776	0.791
CRF	M3 inner	0.452	0.412	0.431
mCRF	M1	0.765	0.731	0.748
ME-BD	M1	0.786	0.734	0.759

Table 3: Official GermEval 2014 evaluation scores [5]

4 Conclusions

In our experiments we could verify that indeed CRFs produce better results compared to an improved ME (see Table 1), but the margin can be minimized by additionally applying further annotators (see Table 3).

We could also verify that it is possible to prune the feature space and thus, reduce resource consumption of NER models significantly to sizes which enable the NER system to be employed directly on mobile devices. Furthermore, the gap to the unrestricted CRF model (1.6%) is relatively small considering the huge amount of saved memory.

References

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Pado. 2014. GermEval 2014 Named Entity Recognition: Companion Paper. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, Hildesheim, Germany.

D. M. Bikel, R. Schwartz, and R. M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, 34(1-3):211–231.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Phd, New York University.

M. Faruqui and S. Padó. 2010. Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, pages 129–133.

J. R. Finkel, T. Grenager, and C. D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of ACL 2005*, pages 363–370.

P. Gamallo and S. Bordag. 2011. Is singular value decomposition useful for word similarity extraction? *Language Resources and Evaluation*, 45(2):95–119.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML 2001*, pages 282–289. Morgan Kaufmann Publishers Inc.

M. Lamar, Y. Maron, M. Johnson, and E. Bienenstock. 2010. SVD and Clustering for Unsupervised POS Tagging. In *Proceedings of ACL 2010*, pages 215–219.

T. R. Leek. 1997. *Information Extraction Using Hidden Markov Models*. Master of science, University of California, San Diego.

E. F. T. K. Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL 2003*, pages 142–147.

M. Tkachenko and A. Simanovsky. 2012. Named Entity Recognition : Exploring Features. In *Proceedings of KONVENS 2012*, pages 118–127.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL 2003*, pages 173–180.

[5] See (Benikova et al., 2014) for metric definitions.