MoSTNER: Morphology-aware Split-Tag German NER with Factorie

Peter Schüller
Computer Engineering Department,
Faculty of Engineering, Marmara University, Istanbul, Turkey
peter.schuller@marmara.edu.tr

Abstract

MoSTNER is a German NER system based on machine learning with log-linear models and morphology-aware features. We use morphological analysis with Morphisto for generating features, moreover we use German Wikipedia as a gazetteer and perform punctuation-aware and morphology-aware page title matching. We use four types of factor graphs where NER labels are single variables or split into prefix (BILOU) and type (PER, LOC, etc.) variables. Our system supports nested NER (two levels), for training we use SampleRank, for prediction Iterated Conditional Modes, the implementation is based on Python and Factorie.

1 Introduction

Various Named Entity Recognition (NER) methods have been developed over time (Nadeau and Sekine, 2007) and currently many state-of-the-art systems rely on variations of Conditional Random Fields (CRF) (Sha and Pereira, 2003), with modifications that step away slightly from the Linear-Chain property, for example Skip-Chains (Sutton and McCallum, 2004), other non-local dependencies (Finkel et al., 2005), and Skip-Grams (Passos et al., 2014). Krishnan and Manning (2006) furthermore described an approach where two layers of CRFs are used to improve predictions of a single level of NER labels.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Page numbers and proceedings footer are added by the organizers. License details: http://creativecommons.org/licenses/by/4.0/

In the GermEval2014 competition for German nested NER several novel challenges needed to be addressed: German capitalization is not a useful feature as in English: adjectives and adverbs derived from names are not capitalized, while all nouns are capitalized; the rich morphology of German creates large noun compounds and makes Gazetteer usage challenging (these only contain the citation form); and nested NER is more challenging than single-level NER.

We next describe features used in the MoSTNER system, four variations of statistical models (some differ from linear-chain CRF quite much), learning and prediction methods, and performance on the GermEval2014 development set.

2 Features and Gazetteer Matching

We use most of the features that are well-known for English NER: token with simplified digits, POS-tag, shape, 4-letter token prefix and suffix, set of tokens in a left/right window of 1 to 4 tokens, POS-bigrams, and token/POS features shifted up to 2 tokens to left and right. POS-tagging was done with the Stanford tagger (Toutanova et al., 2003), moreover we use similarity-clustering using Clark's (2003) software with 400 clusters and 2 training iterations on 10M sentences (263M tokens) from the SdeWaC (Faaß and Eckart, 2013) corpus.

Novel features we added are the following:

- features based on morphological analysis with Morphisto (Schmid et al., 2004; Zielinski and Simon, 2008),
- German Wikipedia categories based on morphology-aware page title matching, and

- POS-bigram of the tag before and after the current token.

For each token, Morphisto generates a list of analyses that contains a sequence of token parts, analyzed as stems and morphological tags.

For example the token 'Presseberichten' ('news reports') obtains 15 analyses, one of them is 'Presse[NN]Bericht[+NN,Masc,Dat,Pl]'. We reduce these analyses by stripping off gender, case, and number morphological tags, and eliminating duplicates. For the above example this yields three analyses: 'Presse[NN]Bericht[+NN]', 'Presse[NN]be[Pref]richten[V,Suff,+NN]', and 'Presse[NN]berichten[V,Suff,+NN]'. From this reduced set of analyses and tags we create 10 distinct feature sets as follows:

- first/last/all stems of the token,
- all tags of the first/last/all token parts, and
- combinations of first/last stems in the left/right window of four neighbor tokens.

As Gazetteer we use German Wikipedia (dump from 20.3.2014) where we perform matching on page titles and redirection pages. Morphology-awareness is achieved by matching only a part of the input sequence (up to 3 characters from the end) in the Wikipedia database and then verifying all results against a regular expression built from the input that allows certain changes to the input sequence with the goal of transforming the input into its citation form: e.g., by stripping a final 's' we can transform genitive 'Maria-s' into nominative 'Maria', or by stripping final 'en' and allowing a Vovel to be added we can allow 'Kont-en' (accounts) and 'Vill-en' (villas) to match their citation forms 'Konto' and 'Villa', respectively.

From those Wikipedia page titles that match the training corpus, we select 1016 page categories (all that are found at least 10 times). If a Wikipedia page title matches a given sequence of tokens in the input, we generate features corresponding to each selected category as follows:

- each token obtains a feature containing the category;
- each token obtains a feature containing the category and a corresponding BILU tag, depending whether it is the first, interior, last, or unique token matching the page title.

This is also done for all subtokens of a token that can be split on a '-' symbol, e.g., 'EU-Minister'.

Stack CRF model		Single split-tag model	
Factors	# Weights	Factors	# Weights
		$bias^p_y$, $bias^p_z$	$5{+}5{=}10$
$bias$	$2{\cdot}49 = 98$	$bias^t_y$, $bias^t_z$	$13{+}13{=}26$
$-$	$-$	$stack^p$, $stack^t$	$5^2{+}13^2{=}194$
		$mark^p_y$, $mark^p_z$	$5^2{\cdot}2{=}50$
		$mark^t_y$, $mark^t_z$	$13^2{\cdot}2{=}338$
		$combo_y$	$5^2{\cdot}13^2{=}4225$
$mark$	$2{\cdot}49^2{=}4802$	$combo_z$	$5^2{\cdot}13^2{=}4225$
		$feat^p_y$, $feat^p_z$	$5{\cdot}2{\cdot}\lvert F\rvert$
$feat$	$2{\cdot}49{\cdot}\lvert F\rvert$	$feat^t_y$, $feat^t_z$	$13{\cdot}2{\cdot}\lvert F\rvert$
total	$4900{+}98{\cdot}\lvert F\rvert$	total	$9068{+}36{\cdot}\lvert F\rvert$

Table 1: Factors and number of weights in (i) a stack of 2 CRF models, and (ii) in a single model with split tags. Note that we use BILOU (5 possibilities) and GermEval uses 12 different NER types (PER, LOC, OTH, ORG, four derived, and four part subtypes).

Additionally we create the same features using a partial matching where any last three characters of the token sequence or the page title can be different. Partial matches are a separate feature set to allow the learning to assign different levels of confidence to partial and exact matches.

Moreover, if there is a pair of punctuation signs (e.g., between '"' and '"', between '(' and ')', and between '-' and '-') around 2 to 4 tokens, we copy all non-BILU Gazetteer features from first and last token to these tokens.

3 Factor Graph Layout(s)

We experimented with four statistical models. MoSTNER is implemented using Python (feature generation) and Factorie (training and prediction of statistical models). We train and predict using BILOU as suggested in (Ratinov, 2012). Figure 1 shows a Linear-Chain CRF for one level of NER labeling on the left side, and a model for labeling two levels of NER with split-tag variables on the right side. The most important characteristic of the split-tag model is that it splits each NER tag (e.g., 'B-POSderiv') into two variables: the BILOU prefix (e.g., 'B') and the NER type (e.g., 'POSderiv'). The idea is to connect the concerns of predicting BILOU with the concern of predicting NER types only where necessary.

Details of the model are as follows: prefixes

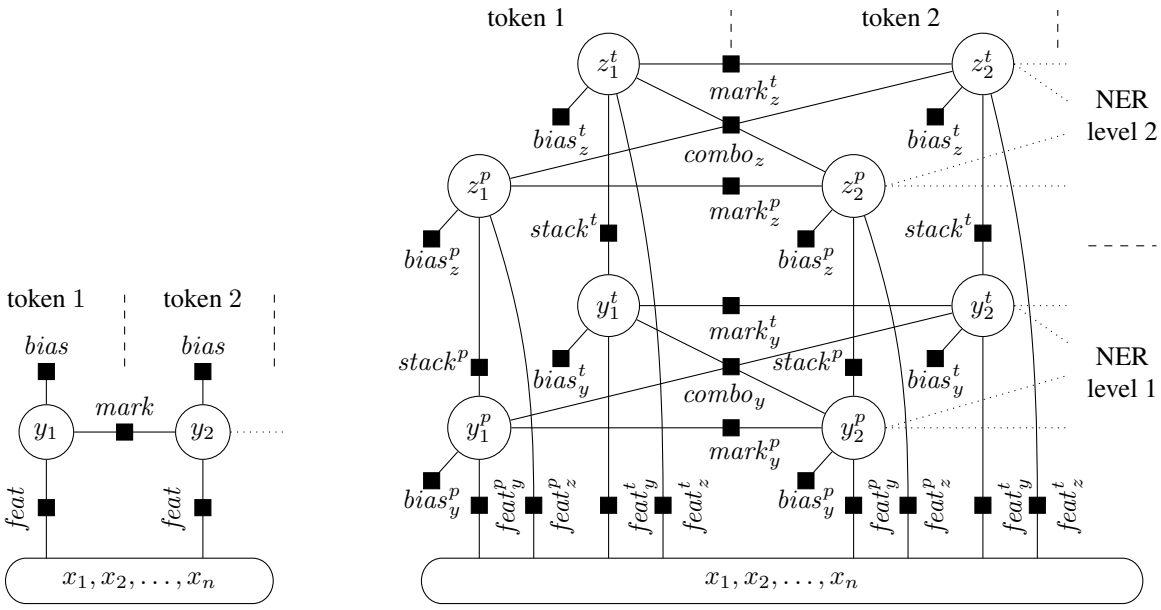

Figure 1: Linear-Chain CRF (left) and single-model split-tag factor graph (right).

and types obtain biases (factor $bias$) and they are connected via Markov-chains within their respective layers (factors $mark$), moreover the two NER levels are connected via factors $stack$ and each level has separate training weights (e.g., factors $feat_p^y$ vs $feat_p^z$ for prefix features). The only factor that relates prefix and type (for span consistency) is $combo$, and this factor does not connect levels.

As shown in Table 1, splitting the tags has the consequence that our single split-tag model obtains fewer weights to train compared with two stacked Linear-Chain CRFs that predict each level separately with one variable per tag. (This is due to the usually high number of features $|F|$.)

4 Experiments

We experimented with four types of models: stacking two Linear-Chain CRFs (Fig. 1 left), using a single split-tag model (Fig. 1 right), stacking two split-tag models (not depicted, imagine stacking two models containing only NER level 1 of Fig. 1 on the right) and using a single model that includes two CRFs stacked on top of each other (not depicted, imagine Fi.e 1 on the right without split tags). For stack models we first train a model to predict the first (outer) NER layer and then a model for the second layer that obtains the first level's predictions as additional features.

For the Linear Chain model we used Viterbi (exact inference) and update weights using the Adaptive Subgradient method by Duchi et al. (2010). The other three models contain cycles, hence exact training and inference methods are not available. We therefore train with SampleRank (Wick et al., 2009) using Gibbs Sampling and a temperature of 0.0001,[1] we update weights using MIRA (Crammer and Singer, 2003). For prediction we use Iterated Conditional Modes (2 iterations) (Besag, 1986). Other learning methods and parameters performed slightly worse.

Model	Notes	Level 1 P-R-F1(%)	Both Levels P-R-F1(%)
Stack-single	Fig. 1 left	76-71-**73.5**	75-69-**72.1**
Stack-split	not depicted	71-67-68.8	71-65-67.7
Single-single	not depicted	74-72-72.8	73-70-71.6
Single-split	Fig. 1 right	73-71-71.9	72-69-70.4

Table 2: GermEval2014 development set performance comparison (official, strict metric). Stack models consist of two separate models, one for each NER level, while single models predict both levels together.

5 Related Work and Conclusion

Faruqui and Padó (2010) described a German NER system with distributed similarity cluster-

[1] For more greedy training (thanks to Michael Wick).

ing and morphology-based features with a linear-chain CRF. MoSTNER additionally uses morphology for Gazetteer lookup and we experiment with more complex models. We did not consider parsing-based approaches as done by Finkel and Manning (2009) for English nested NER.

Performance of MoSTNER on the GermEval2014 (Benikova et al., 2014) development set is shown in Table 2: results indicate that the simplest solution (two Linear-Chain CRFs, one for each NER level) achieves the best prediction correctness. F1-scores on the test set of GermEval2014 are shown in the following table for all the metrics used in the competition.

Model	run	strict	loose	level 1	level 2
Stack-single	3	71.59	72.26	73.24	47.45
Single-split	2	69.18	70.17	70.59	43.80

Experiments with feature sets show that Morphisto features and partial Wikipedia matches decrease performance of the simple CRF, while they increase performance of other models. We plan to perform future work on these observations and publish the source code of MoSTNER.

Acknowledgments

We are grateful for help received from Factorie mailing list members. This work is supported by the Faculty of Engineering, Marmara University.

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.

Julian Besag. 1986. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society Series B (Methodological)*, 48(3):259–302.

Alexander Clark. 2003. Combining Distributional and Morphological Information for Part of Speech Induction. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 59–66.

Koby Crammer and Yoram Singer. 2003. Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, 3:951–991.

John Duchi, Elad Hazan, and Yoram Singer. 2010. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Technical report, University of California at Berkeley.

Gertrude Faaß and Kerstin Eckart. 2013. SdeWaC - A Corpus of Parsable Sentences from the Web. In *Language Processing and Knowledge in the Web*, pages 61–68.

Manaal Faruqui and Sebastian Padó. 2010. Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. In *Verarbeitung natürlicher Sprache (KONVENS)*.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested Named Entity Recognition. In *EMNLP*, pages 141–150.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL*, pages 363–370.

Vijay Krishnan and Christopher D Manning. 2006. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. In *ACL/COLING*, pages 1121–1128.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon Infused Phrase Embeddings for Named Entity Resolution. In *CoNLL*.

Lev Ratinov. 2012. *Exploiting Knowledge in NLP*. Phd thesis, University of Illinois at Urbana-Champaign.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection. In *LREC*, pages 1263–1266.

Fei Sha and Fernando Pereira. 2003. Shallow Parsing with Conditional Random Fields. In *NAACL-HLT*, pages 134–141.

Charles Sutton and Andrew McCallum. 2004. Collective Segmentation and Labeling of Distant Entities in Information Extraction. In *ICML Workshop on Statistical Relational Learning*.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *NAACL-HLT*, pages 252–259.

Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. 2009. SampleRank: Learning Preferences from Atomic Gradients. In *NIPS Workshop on Advances in Ranking*, pages 69–73.

Andrea Zielinski and Christian Simon. 2008. Morphisto - An Open Source Morphological Analyzer for German. In *Workshop on Finite-State Methods and Natural Language Processing*, pages 224–231. IOS Press.