A Semi-Markov Structured Support Vector Machine Model for High-Precision Named Entity Recognition

Ravneet Arora, Chen-Tse Tsai, Ketevan Tsereteli, Prabhanjan Kambadur, Yi Yang*
Bloomberg L.P., *ASAPP Inc.
{rarora62,ctsai54,ktsereteli1,pkambadur}@bloomberg.net,
*yyang@asapp.com

Abstract

Named entity recognition (NER) is the backbone of many NLP solutions. F_1 score, the harmonic mean of precision and recall, is often used to select/evaluate the best models. However, when precision needs to be prioritized over recall, a state-of-the-art model might not be the best choice. There is little in the literature that directly addresses training-time modifications to achieve higher precision information extraction. In this paper, we propose a neural semi-Markov structured support vector machine model that controls the precision-recall trade-off by assigning weights to different types of errors in the loss-augmented inference *during training*. The semi-Markov property provides more accurate phrase-level predictions, thereby improving performance. We empirically demonstrate the advantage of our model when high precision is required by comparing against strong baselines based on CRF. In our experiments with the CoNLL 2003 dataset, our model achieves a better precision-recall trade-off at various precision levels.

1 Introduction

Named Entity Recognition (NER) is the task of locating and categorizing phrases into a closed set of classes, such as organizations, people, and locations. NER is an information extraction task that is important for understanding large bodies of text and is an essential component for many natural language processing (NLP) pipelines. The most common evaluation metric for information extraction tasks is F_1, which is the harmonic mean between precision and recall: that is, false positives and false negatives are weighted equally.

In certain real-world applications (e.g., medicine and finance), extracting wrong information is much worse than extracting nothing: hence, in such domains, high precision is emphasized. Trade-offs between precision and recall have been well researched for classification (Joachims, 2005; Jansche, 2005; Cortes and Mohri, 2004). However, barring studies on inference-time heuristics, there is limited work on training precision-oriented sequence tagging models. In this paper, we present a method for training precision-driven NER models.

By defining custom loss objectives for the structured SVM (SSVM) model, we extend cost-sensitive learning (Domingos, 1999; Margineantu, 2001) to sequence tagging problems. A difficulty in applying cost-sensitive learning to NER is that the model needs to operate on segmentations of the input sentence and the labels of the segments. Inspired by semi-Markov CRF (Sarawagi and Cohen, 2005), we propose a semi-Markov SSVM model that scores and labels consecutive tokens together, which allows us to directly interact with the segment-level errors in the precision-beneficial loss of the SSVM model.

We compare our semi-Markov SSVM model with several competitive inference-time baselines that have been proposed for high-precision NER. Our results show that our model outperforms competitive baselines on organization names, and is at least as good as the best inference-time approaches at some precision levels for other NER classes.

2 Related Work

For classification, several papers try to optimize different evaluation metrics directly. Joachims (2005) proposes an SSVM model for optimizing multivariate performance measures of binary classification tasks. F_β is one of the metrics in their example. Similarly, Jansche (2005) maximizes expected F-measure, Cortes and Mohri (2004) and Narasimhan and Agarwal (2013) optimize AUC

*Work conducted while working at Bloomberg L.P.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5862–5866
Florence, Italy, July 28 - August 2, 2019.

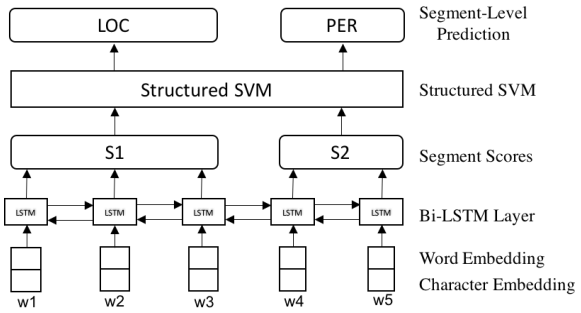

Figure 1: Semi-Markov SSVM model architecture.

and partial AUC, respectively. However, these cannot be directly applied to sequence tagging as labels are assigned at the token or segment level.

Cost-sensitive classification (Domingos, 1999; Margineantu, 2001; Elkan, 2001; Zadrozny et al., 2003) is another body of work where different mis-classification errors have different costs and one attempts to minimize the total cost that a model incurs on the test data. Our approach uses similar ideas – we make the costs of false positive prediction higher than the false-negative costs – and therefore can be viewed as a cost-sensitive model for sequence tagging problems.

For sequence tagging problems, inference-time heuristics for tuning the precision-recall trade-off for information extraction models have been proposed. Culotta and McCallum (2004) calculate confidence scores of the extracted phrases from a CRF model: these scores are used for sorting and filtering extractions. Similarly, Carpenter (2007) computes phrase-level conditional probabilities from an HMM model, and try to increase the recall of gene name extraction by lowering the threshold on these probabilities. Given a trained CRF model, Minkov et al. (2006) hyper-tune the weight for the feature which indicates the token is not a named entity. Changing this weight could encourage or discourage the CRF decoding process to extract entities. We compare our model with these inference-time approaches.

3 Models

We adopt the BiLSTM-CNNs architecture (Ma and Hovy, 2016) to extract features from a sequence of words for all models in this paper. [1] Each word is passed through character-level CNN, and the result is concatenated with Glove word embedding (Pennington et al., 2014) to form the input of Bi-directional LSTM. To map the word representation obtained from BiLSTM into k (label) dimensions, one layer of feed-forward neural network is applied.

At the output layer, instead of using a CRF (Lafferty et al., 2001) to capture the output label dependencies, we use the SSVM objective (Tsochantaridis et al., 2004). While CRFs have consistently given state-of-the-art NER results, their objective function is difficult to directly modify for high-precision extraction. Hence, we select the SSVM formulation as it allows us to directly modify the loss function for high precision. Given training sequences $(\mathbf{x}_i, \mathbf{y}_i), i = 1 \ldots m$, the loss function for SSVM is:

$$\sum_{i=1}^{m} \text{argmax}_{\mathbf{y} \in \mathbf{Y}_{\mathbf{x}_i}} (\Delta(\mathbf{y}_i, \mathbf{y}) + s(\mathbf{y}, \mathbf{x}_i) - s(\mathbf{y}_i, \mathbf{x}_i)),$$

where Δ is the Hamming loss between two sequences, $\mathbf{Y}_{\mathbf{x}_i}$ contains all possible label assignments for the sentence $\mathbf{x}_i$, and s is the decoding score between input sentence $\mathbf{x}$ and label sequence $\mathbf{y}$.

3.1 High-Precision SSVM

Without modifications, the SSVM performs similar to the CRF. However, the presence of $\Delta(\mathbf{y}_i, \mathbf{y})$ in the SSVM loss allows us to design custom loss functions for high precision NER. No inference-time changes are introduced.

Class-specific Token-level Loss The first modification we make is to pick a target entity class and modify $\Delta(\mathbf{y}_i, \mathbf{y})$ to have word-wise loss of ℓ_{tgt} for false positives on the target class and loss of $\ell_{\tilde{tgt}}$ for false positives on other classes. That is, let $\mathbf{y}_i^j$ be j-th element of sequence $\mathbf{y}_i$, we define $\Delta(\mathbf{y}_i, \mathbf{y}) = \sum_j w_j$, where

$$w_j = \begin{cases} 0, & \text{if } \mathbf{y}_i^j = \mathbf{y}^j \\ \ell_{tgt}, & \text{if } \mathbf{y}_i^j \neq \mathbf{y}^j \text{ and } \mathbf{y}^j = \text{target class} \\ \ell_{\tilde{tgt}}, & \text{if } \mathbf{y}_i^j \neq \mathbf{y}^j \text{ and } \mathbf{y}^j \neq \text{target class} \end{cases}$$

Note that the target class in the above equation contains all the labels related to the target entity type; that is, if the target class is ORG, we consider B-ORG and I-ORG to be the related labels. Typically $\ell_{tgt} \gg \ell_{\tilde{tgt}}$ so that the false positives on the target class will generate more loss, thereby discouraging the model from making such decisions. Both ℓ_{tgt} and $\ell_{\tilde{tgt}}$ are determined through

[1] Our implementation is based on NCRF++ (Yang and Zhang, 2018).

hyper-parameter tuning. Setting $\ell_{tgt} = \ell_{\tilde{tgt}} = 1$ falls back to the standard Hamming loss.

Semi-Markov SSVM A problem with token-level loss is that it does not always reflect phrase-level errors accurately; it may over generate loss since a phrase could consist of multiple tokens. It is unclear how individual token false positives contribute to phrase-level false positives.

Therefore, we try a semi-Markov variation of the SSVM following (Sarawagi and Cohen, 2005). The semi-Markov formulation groups consecutive tokens into segments. Whole segments are considered as a single unit and only transitions between segments are modeled. We ignore all intra-segment transition probabilities, effectively collapsing the number of labels to 5 (*ORG, PER, LOC, MISC, O* instead of the BIO labelling scheme for CoNLL data). The scores of each segment are obtained by summing up the word-level class scores of words present in the segment (Ye and Ling, 2018). We restrict segments to be ≤ 7 tokens long, and we do not use any additional segment level features. During decoding, all possible segmentations of a sentence (≤ 7) will be considered. The architecture of our BiLSTM semi-Markov SSVM model is shown in Figure 1.

To tune the semi-Markov SSVM model to high precision for a specific class, a segment will contribute ℓ_{tgt} to the loss if it is predicted as the target class and this segment does not exist in the gold segmentation. Other types of errors in the prediction have a loss of $\ell_{\tilde{tgt}}$. This is similar to the class-specific loss used on the token-level in the SSVM formulation. In our experiments, we refer to the token-level model simply as SSVM, and the segment-level model as semi-Markov SSVM.

4 Results

All experiments were conducted on the CoNLL 2003 English dataset. We first show the performance of CRF, SSVM, and semi-Markov SSVM models without tuning for high precision in Table 1. We see that all three models perform similarly, with CRF being slightly better. These numbers are the starting points for the rest of the experiments. We compare the proposed models with the following inference-time baselines:[2]

[2]Results of Minkov et al. (2006) are given in the Appendix as the performance is worse than the other methods.

		ORG	PER	LOC	MISC	ALL
CRF	P.	89.5	96.3	91.8	81.1	91.06
	R.	87.7	95.4	93.8	81.3	90.88
	F1	88.6	95.8	92.8	81.2	90.97
SSVM	P.	90.0	95.7	91.0	80.4	90.75
	R.	87.7	95.5	93.7	80.5	90.79
	F1	88.8	95.6	92.4	80.4	90.77
Semi. SSVM	P.	89.3	96.0	92.3	80.1	90.92
	R.	87.2	95.2	93.2	81.9	90.60
	F1	88.2	95.6	92.8	81.0	90.76

Table 1: Performance of the baseline and proposed models without tuning for high precision. These numbers are on the CoNLL 2003 English test set. The development set is not included in training.

ORG (Precision: 94.5)			
Ment. Length	1(65.1%)	2(24.3%)	≥ 3(10.6%)
Thres. CRF	**84.94**	78.16	75.57
Semi. SSVM	84.57	**80.40**	**83.52**
LOC (Precision: 95.5)			
Ment. Length	1(86.1%)	2(12.4%)	≥ 3(1.5%)
Thres. CRF	**92.90**	90.82	60.00
Semi. SSVM	92.06	**91.79**	**64.00**
PER (Precision: 97.9)			
Ment. Length	1(32.8%)	2(63.0%)	≥ 3(4.2%)
Thres. CRF	**81.73**	97.74	91.18
Semi. SSVM	81.54	**99.02**	**95.59**

Table 2: Recall of the thresholded CRF and semi-Markov SSVM for different mention lengths at the same precision level. The chosen precision levels are listed right next to the entity types. The percentages in parenthesis are of the gold mentions.

Thresholded CRF We compute the probability of each extracted phrase by Constrained Forward-Backward algorithm (Culotta and McCallum, 2004). An extraction is dropped if its phrase probability is lower than a given threshold, a tunable hyper-parameter.

Bootstrap CRF By generating bootstrap samples of the CoNLL training set, we generate 100 BiLSTM CRF models. To increase precision over a single CRF, we decode each sentence with each of the 100 models and compute the votes for each proposed named entity. The threshold (percent of votes) for a candidate entity is hyper-tuned.

Using the dev set, we tune the hyper-parameters of each model at which the desired precision is achieved. For our proposed SSVM-based mod-

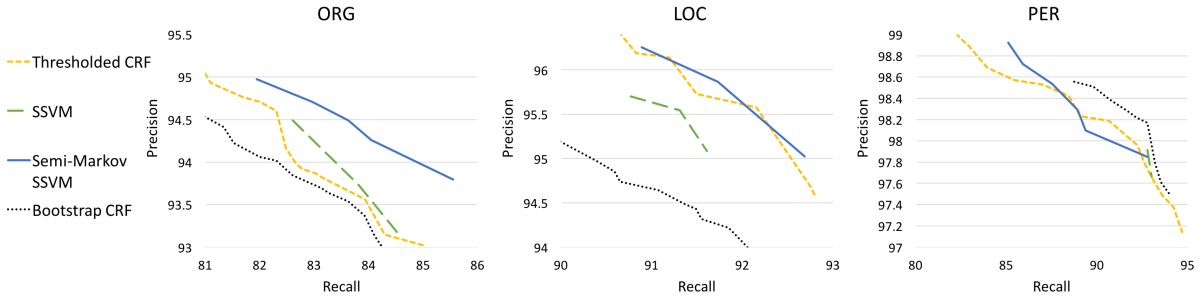

Figure 2: Precision-recall trade-off of the proposed SSVM model versus baselines: semi-Markov SSVM outperforms all models for ORG, is on par with Thresholded CRF for LOC, and is competitive for the PER class. The detailed numbers are listed in the Appendix.

els, the hyper-parameters are ℓ_{tgt} and $\ell_{\tilde{tgt}}$.[3] To speed up training, we initialize the parameters of the entire model (neural network and SSVM) using a pre-trained model with $\ell_{tgt} = 1$, $\ell_{\tilde{tgt}} = 1$, and train further for 20 epochs.

We set several precision levels from 90 to 100. For each precision level, we choose the hyper-parameters which have precision higher than the target precision level and obtain the maximum F_1 score on the dev set, and report the corresponding test performance. The results are shown in Figure 2. Threshold CRF can achieve a wider range of precision than SSVM-based models. In this figure, we only focus on the range which SSVM-based models can achieve.

We can see that semi-Markov SSVM clearly outperforms all the other models for ORG, is on par with Thresholded CRF for LOC, and has some strong points in the high precision region for PER. The good performance on ORG is consistent with the observation in Ye and Ling (2018) that semi-Markov models have advantages in longer phrases because labels are assigned at the segment level directly. Since longer mentions tend to have a smaller phrase probability and the length of ORG mentions varies more than the length of the other two types, Thresholded CRF is less robust for ORG. The token-based SSVM is consistently worse than semi-Markov SSVM and fails to achieve higher precision, especially for PER. This shows that the semi-Markov property penalizes false positives at the phrase-level more accurately. Bootstrap CRF does not perform well for ORG and LOC, but is pretty strong for PER at some precision levels. We believe higher performance of bootstrap CRF on PER class comes from the fact that the baseline CRF model itself achieves very high precision for this class, which allows bootstrapping technique reduce the variance on predictions accurately. This makes bootstrapping approach more promising to situations where models have already achieved very high precision.

4.1 Error Analysis

We perform error analysis for the two main methods: Thresholded CRF and semi-Markov SSVM. We pick model settings such that both models achieve the same precision level (ORG:94.5 PER:97.9 LOC:95.5) for a given class. Table 2 illustrates the recall values achieved by these models for different entity mention lengths. We can see that semi-Markov SSVM clearly outperforms Thresholded CRF on multi-token mentions, especially for long organization names. The high percentage of long mentions in ORG explains semi-Markov SSVM's superior performance in Figure 2. However, we also see that semi-Markov SSVM produces more "larger predicted span" errors. Therefore the recall of unit-length mentions is lower than Thresholded CRF. This we believe is a side effect of semi-Markov models being more willing to predict longer length segments.

These two methods can be applied together to achieve even better results. For example, thresholding and bootstrap techniques can be applied to semi-Markov SSVM models as well. In this work, we focus on showing the performance of individual approaches.

Another question is what types of errors are reduced when tuning towards precision? We find that precision tuning reduces all error types, but especially the MISC type errors for all 3 classes (i.e., MISC being classified as one of the other 3 classes).

[3] ℓ_{tgt} is searched in the range between 1 and 5, and $\ell_{\tilde{tgt}}$ is between 0.0001 and 0.1.

5 Conclusion

We proposed a semi-Markov SSVM model for high-precision NER. To our best knowledge, it is the first training-time model for high precision structured prediction. Experiment results show that our model performs better than inference-time approaches at several precision levels, especially for longer mentions. The proposed model offers promising future extensions in terms of directly optimizing other metrics such as Recall and F_β. This work also opens up a range of questions from modeling to evaluation methodology.

References

Bob Carpenter. 2007. Lingpipe for 99.99% recall of gene mentions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 307–309.

Corinna Cortes and Mehryar Mohri. 2004. AUC optimization vs. error rate minimization. In *Advances in neural information processing systems*, pages 313–320.

Aron Culotta and Andrew McCallum. 2004. Confidence estimation for information extraction. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 109–112.

Pedro Domingos. 1999. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164.

Charles Elkan. 2001. The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference on Artificial intelligence*, pages 973–978.

Martin Jansche. 2005. Maximum expected F-measure training of logistic regression models. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 692–699.

Thorsten Joachims. 2005. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pages 377–384.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1064–1074.

Dragos Dorin Margineantu. 2001. Methods for cost-sensitive learning. *PhD Thesis, Oregon State University*.

Einat Minkov, Richard C Wang, Anthony Tomasic, and William W Cohen. 2006. NER systems that suit user's preferences: adjusting the recall-precision trade-off for entity extraction. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 93–96.

Harikrishna Narasimhan and Shivani Agarwal. 2013. SVM pAUC tight: a new support vector method for optimizing partial auc based on a tight convex upper bound. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–175.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Advances in neural information processing systems*, pages 1185–1192.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the international conference on Machine learning*, page 104.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Bianca Zadrozny, John Langford, and Naoki Abe. 2003. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 435–442.