Hybrid semi-Markov CRF for Neural Sequence Labeling

Zhi-Xiu Ye
University of Science and Technology of China
zxye@mail.ustc.edu.cn

Zhen-Hua Ling
University of Science and Technology of China
zhling@ustc.edu.cn

Abstract

This paper proposes hybrid semi-Markov conditional random fields (SCRFs) for neural sequence labeling in natural language processing. Based on conventional conditional random fields (CRFs), SCRFs have been designed for the tasks of assigning labels to segments by extracting features from and describing transitions between segments instead of words. In this paper, we improve the existing SCRF methods by employing word-level and segment-level information simultaneously. First, word-level labels are utilized to derive the segment scores in SCRFs. Second, a CRF output layer and an SCRF output layer are integrated into an unified neural network and trained jointly. Experimental results on CoNLL 2003 named entity recognition (NER) shared task show that our model achieves state-of-the-art performance when no external knowledge is used[1].

1 Introduction

Sequence labeling, such as part-of-speech (POS) tagging, chunking, and named entity recognition (NER), is a category of fundamental tasks in natural language processing (NLP). Conditional random fields (CRFs) (Lafferty et al., 2001), as probabilistic undirected graphical models, have been widely applied to the sequence labeling tasks considering that they are able to describe the dependencies between adjacent word-level labels and to avoid illegal label combination (e.g., I-ORG can't follow B-LOC in the NER tasks using the BIOES tagging scheme). Original CRFs utilize hand-crafted features which increases the difficulty of performance tuning and domain adaptation. In recent years, neural networks with distributed word representations (i.e., word embeddings) (Mikolov et al., 2013; Pennington et al., 2014) have been introduced to calculate word scores automatically for CRFs (Chiu and Nichols, 2016; Huang et al., 2015).

On the other hand, semi-Markov conditional random fields (SCRFs) (Sarawagi and Cohen, 2005) have been proposed for the tasks of assigning labels to the segments of input sequences, e.g., NER. Different from CRFs, SCRFs adopt segments instead of words as the basic units for feature extraction and transition modeling. The word-level transitions within a segment are usually ignored. Some variations of SCRFs have also been studied. For example, Andrew (2006) extracted segment-level features by combining hand-crafted CRF features and modeled the Markov property between words instead of segments in SCRFs. With the development of deep learning, some models of combining neural networks and SCRFs have also been studied. Zhuo et al. (2016) and Kong et al. (2015) employed gated recursive convolutional neural networks (grConvs) and segmental recurrent neural networks (SRNNs) to calculate segment scores for SCRFs respectively.

All these existing neural sequence labeling methods using SCRFs only adopted segment-level labels for score calculation and model training. In this paper, we suppose that word-level labels can also contribute to the building of SCRFs and thus design a hybrid SCRF (HSCRF) architecture for neural sequence labeling. In an HSCRF, word-level labels are utilized to derive the segment scores. Further, a CRF output layer and an HSCRF output layer are integrated into a unified neural network and trained jointly. We evaluate our model on CoNLL 2003 English NER task (Sang and Meulder, 2003) and achieve

[1]The code of our models is available at http://github.com/ZhixiuYe/HSCRF-pytorch

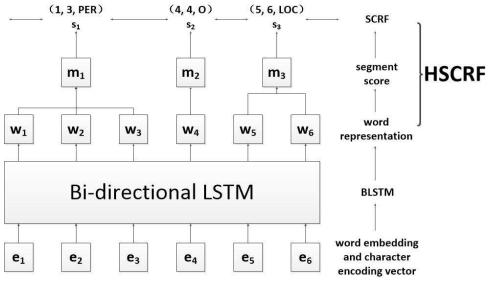

Figure 1: The diagram of a neural network with an HSCRF output layer for sequence labeling.

state-of-the-art performance when no external knowledge is used.

In summary, the contributions of this paper are: (1) we propose the HSCRF architecture which employs both word-level and segment-level labels for segment score calculation. (2) we propose a joint CRF-HSCRF training framework and a naive joint decoding algorithm for neural sequence labeling. (3) we achieve state-of-the-art performance in CoNLL 2003 NER shared task.

2 Methods

2.1 Hybrid semi-Markov CRFs

Let $\mathbf{s} = \{s_1, s_2, ..., s_p\}$ denote the segmentation of an input sentence $\mathbf{x} = \{x_1, ..., x_n\}$ and $\mathbf{w} = \{w_1, ..., w_n\}$ denote the sequence of word representations of $\mathbf{x}$ derived by a neural network as shown in Fig. 1. Each segment $s_i = (b_i, e_i, l_i)$, $0 \leq i \leq p$, is a triplet of a begin word index b_i, an end word index e_i and a segment-level label l_i, where $b_1 = 1$, $e_p = |\mathbf{x}|$, $b_{i+1} = e_i + 1$, $0 \leq e_i - b_i < L$, and L is the upperbound of the length of s_i. Correspondingly, let $\mathbf{y} = \{y_1, ..., y_n\}$ denote the word-level labels of $\mathbf{x}$. For example, if a sentence $\mathbf{x}$ in NER task is "Barack Hussein Obama and Natasha Obama", we have the corresponding $\mathbf{s} = ((1, 3, PER), (4, 4, O), (5, 6, PER))$ and $\mathbf{y}$ = (B-PER, I-PER, E-PER, O, B-PER, E-PER).

Similar to conventional SCRFs (Sarawagi and Cohen, 2005), the probability of a segmentation $\hat{\mathbf{s}}$ in an HSCRF is defined as

$$\mathrm{p}(\hat{\mathbf{s}}|\mathbf{w}) = \frac{\mathrm{score}(\hat{\mathbf{s}}, \mathbf{w})}{\sum_{\mathbf{s}' \in \mathbf{S}} \mathrm{score}(\mathbf{s}', \mathbf{w})}, \quad (1)$$

where $\mathbf{S}$ contains all possible segmentations and

$$\mathrm{score}(\mathbf{s}, \mathbf{w}) = \prod_{i=1}^{|\mathbf{s}|} \psi(l_{i-1}, l_i, \mathbf{w}, b_i, e_i). \quad (2)$$

Here, $\psi(l_{i-1}, l_i, \mathbf{w}, b_i, e_i) = exp\{m_i + b_{l_{i-1}, l_i}\}$, where $m_i = \varphi_{\mathrm{h}}(l_i, \mathbf{w}, b_i, e_i)$ is the segment score and $b_{i,j}$ is the segment-level transition parameter from class i to class j.

Different from existing methods of utilizing SCRFs in neural sequence labeling (Zhuo et al., 2016; Kong et al., 2015) , the segment score in an HSCRF is calculated using word-level labels as

$$m_i = \sum_{k=b_i}^{e_i} \varphi_{\mathrm{c}}(y_k, \mathbf{w'}_k) = \sum_{k=b_i}^{e_i} \mathbf{a}_{y_k}^{\top} \mathbf{w'}_k, \quad (3)$$

where $\mathbf{w'}_k$ is the feature vector of the k-th word, $\varphi_{\mathrm{c}}(y_k, \mathbf{w'}_k)$ calculates the score of the k-th word being classified into word-level class y_k, and $\mathbf{a}_{y_k}$ is a weight parameter vector corresponding to class y_k. For each word, $\mathbf{w'}_k$ is composed of word representation $\mathbf{w}_k$ and another two segment-level descriptions, i.e., (1) $\mathbf{w}_{e_i} - \mathbf{w}_{b_i}$ which is derived based on the assumption that word representations in the same segment (e.g., "Barack Obama") are closer to each other than otherwise (e.g., "Obama is"), and (2) $\boldsymbol{\phi}(k - b_i + 1)$ which is the embedding vector of the word index in a segment. Finally, we have $\mathbf{w}'_k = [\mathbf{w}_k; \mathbf{w}_{e_i} - \mathbf{w}_{b_i}; \boldsymbol{\phi}(k - b_i + 1)]$, where $b_i \leq k \leq e_i$ and $[;;]$ is a vector concatenation operation.

The training and decoding criteria of conventional SCRFs (Sarawagi and Cohen, 2005) are followed. The negative log-likelihood (NLL), i.e., $-log\mathrm{p}(\hat{\mathbf{s}}|\mathbf{w})$, is minimized to estimate the parameters of the HSCRF layer and the lower neural network layers that derive word representations. For decoding, the Viterbi algorithm is employed to obtain the optimal segmentation as

$$\mathbf{s}^* = \underbrace{\mathrm{argmax}}_{\mathbf{s}' \in \mathbf{S}} log\mathrm{p}(\mathbf{s}'|\mathbf{m}), \quad (4)$$

where $\mathbf{S}$ contains all legitimate segmentations.

2.2 Jointly training and decoding using CRFs and HSCRFs

To further investigate the effects of word-level labels on the training of SCRFs, we integrate a CRF output layer and a HSCRF output layer into an unified neural network and train them jointly. These two output layers share the same sequence of word representations $\mathbf{w}$ which are extracted by lower neural network layers. Given both word-level and segment-level ground truth labels of training sentences, the model parameters

are optimized by minimizing the summation of the loss functions of the CRF layer and the HSCRF layer with equal weights.

At decoding time, two label sequences, i.e., $\mathbf{s}_c$ and $\mathbf{s}_h$, for an input sentence can be obtained using the CRF output layer and the HSCRF output layer respectively. A naive joint decoding algorithm is also designed to make a selection between them. Assume the NLLs of measuring $\mathbf{s}_c$ and $\mathbf{s}_h$ using the CRF and HSCRF layers are NLL_c and NLL_h respectively. Then, we exchange the models and measure the NLLs of $\mathbf{s}_c$ and $\mathbf{s}_h$ by HSCRF and CRF and obtain another two values $NLL_{c\_by\_h}$ and $NLL_{h\_by\_c}$. We just naively assign the summation of NLL_c and $NLL_{c\_by\_h}$ to $\mathbf{s}_c$, and the summation of NLL_h and $NLL_{h\_by\_c}$ to $\mathbf{s}_h$. Finally, we choose the one between $\mathbf{s}_c$ and $\mathbf{s}_h$ with lower NLL sum as the final result.

3 Experiments

3.1 Dataset

We evaluated our model on the CoNLL 2003 English NER dataset (Sang and Meulder, 2003). This dataset contained four labels of named entities (PER, LOC, ORG and MISC) and label O for others. The existing separation of training, development and test sets was followed in our experiments. We adopted the same word-level tagging scheme as the one used in Liu et al. (2018) (e.g., BIOES instead of BIO). For better computation efficiency, the max segment length L introduced in Section 2.1 was set to 6, which pruned less than 0.5% training sentences for building SCRFs and had no effect on the development and test sets.

3.2 Implementation

As shown in Fig. 1, the GloVe (Pennington et al., 2014) word embedding and the character encoding vector of each word in the input sentence were concatenated and fed into a bi-directional LSTM to obtain the sequence of word representations $\mathbf{w}$. Two character encoding models, LM-BLSTM (Liu et al., 2018) and CNN-BLSTM (Ma and Hovy, 2016), were adopted in our experiments. Regarding with the top classification layer, we compared our proposed HSCRF with conventional word-level CRF and grSemi-CRF (GSCRF) (Zhuo et al., 2016), which was an SCRF using only segment-level information. The descriptions of the models built in our experiments are summarized in Table 1.

For a fair comparison, we implemented all models in the same framework using PyTorch library[2]. The hyper-parameters of the models are shown in Table 2 and they were selected according to the two baseline methods without fine-tuning. Each model in Table 1 was estimated 10 times and its mean and standard deviation of F1 score were reported considering the influence of randomness and the weak correlation between development set and test set in this task (Reimers and Gurevych, 2017).

3.3 Results

Table 1 lists the F1 score results of all built models on CoNLL 2003 NER task. Comparing model 3 with model 1/2 and model 9 with model 7/8, we can see that HSCRF performed better than CRF and GSCRF. The superiorities were significant since the p-values of t-test were smaller than 0.01. This implies the benefits of utilizing word-level labels when deriving segment scores in SCRFs. Comparing model 1 with model 4, 3 with 5, 7 with 10, and 9 with 11, we can see that the jointly training method introduced in Section 2.2 improved the performance of CRF and HSCRF significantly ($p < 0.01$ in all these four pairs). This may be attributed to that jointly training generates better word representations that can be shared by both CRF and HSCRF decoding layers. Finally, comparing model 6 with model 4/5 and model 12 with model 10/11, we can see the effectiveness of the jointly decoding algorithm introduced in Section 2.2 on improving F1 scores ($p < 0.01$ in all these four pairs). The LM-BLSTM-JNT model with jointly decoding achieved the highest F1 score among all these built models.

3.4 Comparison with existing work

Table 3 shows some recent results[3] on the CoNLL 2003 English NER task. For the convenience of comparison, we also listed the maximum F1 scores among 10 repetitions when building our models. The maximum F1 score of our re-implemented CNN-BLSTM-CRF model was slightly worse than the one originally reported in

[2] http://pytorch.org/

[3] It should be noticed that the results of Liu et al. (2018) were inconsistent with the original ones reported in their paper. According to its first author's GitHub page (https://github.com/LiyuanLucasLiu/LM-LSTM-CRF), the originally reported results had errors due to some bugs. Here, we report the results after the bugs got fixed.

No.	Model Name	Word Representation	Top Layer	Decoding Layer	F1 Score (±std)
1	CNN-BLSTM-CRF	CNN-BLSTM	CRF	CRF	90.92 ± 0.08
2	CNN-BLSTM-GSCRF	CNN-BLSTM	GSCRF	GSCRF	90.96 ± 0.12
3	CNN-BLSTM-HSCRF	CNN-BLSTM	HSCRF	HSCRF	91.10 ± 0.12
4	CNN-BLSTM-JNT(CRF)	CNN-BLSTM	CRF+HSCRF	CRF	91.08 ± 0.12
5	CNN-BLSTM-JNT(HSCRF)	CNN-BLSTM	CRF+HSCRF	HSCRF	91.20 ± 0.10
6	CNN-BLSTM-JNT(JNT)	CNN-BLSTM	CRF+HSCRF	CRF+HSCRF	91.26 ± 0.10
7	LM-BLSTM-CRF	LM-BLSTM	CRF	CRF	91.17 ± 0.11
8	LM-BLSTM-GSCRF	LM-BLSTM	GSCRF	GSCRF	91.06 ± 0.05
9	LM-BLSTM-HSCRF	LM-BLSTM	HSCRF	HSCRF	91.27 ± 0.08
10	LM-BLSTM-JNT(CRF)	LM-BLSTM	CRF+HSCRF	CRF	91.24 ± 0.07
11	LM-BLSTM-JNT(HSCRF)	LM-BLSTM	CRF+HSCRF	HSCRF	91.34 ± 0.10
12	LM-BLSTM-JNT(JNT)	LM-BLSTM	CRF+HSCRF	CRF+HSCRF	91.38 ± 0.10

Table 1: Model descriptions and their performance on CoNLL 2003 NER task.

Component	Parameter	Value
word-level embedding†‡	dimension	100
character-level embedding†‡	dimension	30
character-level LSTM†	depth	1
	hidden size	300
highway network†	layer	1
word-level BLSTM†	depth	1
	hidden size	300
word-level BLSTM‡	depth	1
	hidden size	200
CNN‡	window size	3
	filter number	30
$\phi(\cdot)^{\dagger\ddagger}$	dimension	10
dropout†‡	dropout rate	0.5
optimization†‡	learning rate	0.01
	batch size	10
	strategy	SGD
	gradient clip	5.0
	decay rate	1/(1+0.05t)

Table 2: Hyper-parameters of the models built in our experiments, where † indicates the ones when using LM-BLSTM for deriving word representations and ‡ indicates the ones when using CNN-BLSTM.

Model	Test Set F1 Score	
	Type	Value (±std)
Zhuo et al. (2016)	reported	88.12
Lample et al. (2016)	reported	90.94
Ma and Hovy (2016)	reported	91.21
Rei (2017)	reported	86.26
Liu et al. (2018)	mean	91.24 ± 0.12
	max	91.35
CNN-BLSTM-CRF	mean	90.92 ± 0.08
	max	91.04
LM-BLSTM-CRF	mean	91.17 ± 0.11
	max	91.30
CNN-BLSTM-JNT(JNT)	mean	91.26 ± 0.10
	max	91.41
LM-BLSTM-JNT(JNT)	mean	**91.38± 0.10**
	max	**91.53**
Luo et al. (2015)*	reported	91.2
Chiu and Nichols (2016)*	reported	91.62 ± 0.33
Tran et al. (2017)*	reported	91.66
Peters et al. (2017)*	reported	91.93 ± 0.19
Yang et al. (2017)*	reported	91.26

Table 3: Comparison with existing work on CoNLL 2003 NER task. The models labelled with * utilized external knowledge beside CoNLL 2003 training set and pre-trained word embeddings.

Ma and Hovy (2016), but it was similar to the one reported in Reimers and Gurevych (2017).

In the NER models listed in Table 3, Zhuo et al. (2016) employed some manual features and calculated segment scores by grConv for SCRF. Lample et al. (2016) and Ma and Hovy (2016) constructed character-level encodings using BLSTM and CNN respectively, and concatenated them with word embeddings. Then, the same BLSTM-CRF architecture was adopted in both models. Rei (2017) fed word embeddings into LSTM to obtain the word representations for CRF decoding and to predict the next word simultaneously. Similarly, Liu et al. (2018) input characters into LSTM to predict the next character and to get the character-level encoding for each word.

Some of the models listed in Table 3 utilized external knowledge beside CoNLL 2003 training set and pre-trained word embeddings. Luo et al. (2015) proposed JERL model, which was trained on both NER and entity linking tasks simultaneously. Chiu and Nichols (2016) employed lexicon features from DBpedia (Auer et al., 2007). Tran et al. (2017) and Peters et al. (2017) utilized pre-trained language models from large corpus to model word representations. Yang et al. (2017) utilized transfer learning to obtain shared information from other tasks, such as chunking and POS

No.	Model Name	Entity Length						
		1	2	3	4	5	≥ 6	all
7	LM-BLSTM-CRF	91.68	91.88	82.64	75.81	73.68	72.73	91.17
8	LM-BLSTM-GSCRF	91.57	91.68	83.61	74.32	76.64	73.64	91.06
9	LM-BLSTM-HSCRF	91.65	91.84	82.97	76.20	78.95	74.55	91.27
12	LM-BLSTM-JNT(JNT)	**91.73**	**92.03**	**83.78**	**77.27**	**79.66**	**76.55**	**91.38**

Table 4: Model performance on CoNLL 2003 NER task for entities with different lengths.

tagging, for word representations.

From Table 3, we can see that our CNN-BLSTM-JNT and LM-BLSTM-JNT models with jointly decoding both achieved state-of-the-art F1 scores among all models without using external knowledge. The maximum F1 score achieved by the LM-BLSTM-JNT model was 91.53%.

3.5 Analysis

To better understand the effectiveness of word-level and segment-level labels on the NER task, we evaluated the performance of models 7, 8, 9 and 12 in Table 3 for entities with different lengths. The mean F1 scores of 10 training repetitions are reported in Table 4. Comparing model 7 with model 8, we can see that GSCRF achieved better performance than CRF for long entities (with more than 4 words) but worse for short entities (with less than 3 words). Comparing model 7 with model 9, we can find that HSCRF outperformed CRF for recognizing long entities and meanwhile achieved comparable performance with CRF for short entities.

One possible explanation is that word-level labels may supervise models to learn word-level descriptions which tend to benefit the recognition of short entities. On the other hand, segment-level labels may guide models to capture the descriptions of combining words for whole entities which help to recognize long entities. By utilizing both labels, the LM-BLSTM-HSCRF model can achieve better overall performance of recognizing entities with different lengths. Furthermore, the LM-BLSTM-JNT(JNT) model which adopted jointly training and decoding achieved the best performance among all models shown in Table 4 for all entity lengths.

4 Conclusions

This paper proposes a hybrid semi-Markov conditional random field (HSCRF) architecture for neural sequence labeling, in which word-level labels are utilized to derive the segment scores in SCRFs. Further, the methods of training and decoding CRF and HSCRF output layers jointly are also presented. Experimental results on CoNLL 2003 English NER task demonstrated the effectiveness of the proposed HSCRF model which achieved state-of-the-art performance.

References

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association of Computational Linguistics*, 4:357–370.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2015. Segmental recurrent neural networks. *arXiv preprint arXiv:1511.06018*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270. Association for Computational Linguistics.

L. Liu, J. Shang, F. Xu, X. Ren, H. Gui, J. Peng, and J. Han. 2018. Empower Sequence Labeling with Task-Aware Neural Language Model. In *AAAI*.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888. Association for Computational Linguistics.

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: Long Papers)*, pages 1064–1074. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

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.

Quan Tran, Andrew MacKinlay, and Antonio Jimeno Yepes. 2017. Named entity recognition with stack residual LSTM and trainable bias decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 566–575. Asian Federation of Natural Language Processing.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Jingwei Zhuo, Yong Cao, Jun Zhu, Bo Zhang, and Zaiqing Nie. 2016. Segment-level sequence modeling using gated recursive semi-Markov In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1413–1423. Association for Computational Linguistics.