Multi-channel BiLSTM-CRF Model for Emerging Named Entity Recognition in Social Media

Bill Y. Lin* and **Frank F. Xu*** and **Zhiyi Luo** and **Kenny Q. Zhu**
Shanghai Jiao Tong University
Shanghai, China
{yuchenlin, frankxu, jessherlock}@sjtu.edu.cn, kzhu@cs.sjtu.edu.cn

Abstract

In this paper, we present our multi-channel neural architecture for recognizing emerging named entity in social media messages, which we applied in the *Novel and Emerging Named Entity Recognition* shared task at the EMNLP 2017 Workshop on Noisy User-generated Text (W-NUT). We propose a novel approach, which incorporates comprehensive word representations with multi-channel information and Conditional Random Fields (CRF) into a traditional Bidirectional Long Short-Term Memory (BiLSTM) neural network without using any additional hand-crafted features such as gazetteers. In comparison with other systems participating in the shared task, our system won the 3rd place in terms of the average of two evaluation metrics.

1 Introduction

Named entity recognition (NER) is one of the first and most important steps in Information Extraction pipelines. Generally, it is to identify mentions of entities (persons, locations, organizations, etc.) within unstructured text. However, the diverse and noisy nature of user-generated content as well as the emerging entities with novel surface forms make NER in social media messages more challenging.

The first challenge brought by user-generated content is its unique characteristics: short, noisy and informal. For instance, tweets are typically short since the number of characters is restricted to 140 and people indeed tend to pose short messages even in social media without such restrictions, such as YouTube comments and Reddit. [1] Hence, the contextual information in a sentence is very limited. Apart from that, the use of colloquial language makes it more difficult for existing NER approaches to be reused, which mainly focus on a general domain and formal text (Baldwin et al., 2015; Derczynski et al., 2015).

Another challenge of NER in noisy text is the fact that there are large amounts of emerging named entities and rare surface forms among the user-generated text, which tend to be tougher to detect (Augenstein et al., 2017) and recall thus is a significant problem (Derczynski et al., 2015). By way of example, the surface form "*kktny*", in the tweet "so.. *kktny* in 30 mins?", actually refers to a new TV series called "*Kourtney and Kim Take New York*", which even human experts found hard to recognize. Additionally, it is quite often that netizens mention entities using rare morphs as surface forms. For example, "*black mamba*", the name for a venomous snake, is actually a morph that Kobe Bryant created for himself for his aggressiveness in playing basketball games (Zhang et al., 2015). Such morphs and rare surface forms are also very difficult to detect and classify.

The goal of this paper is to present our system participating in the *Novel and Emerging Named Entity Recognition* shared task at the EMNLP 2017 Workshop on Noisy User-generated Text (W-NUT 2017), which aims for NER in such noisy user-generated text. We investigate a multi-channel BiLSTM-CRF neural network model in our participating system, which is described in Section 3. The details of our implementation are in presented in Section 4, where we also present some conclusion from our experiments.

* The two authors made equal contributions.

[1] The average length of the sentences in this shared task is about 20 tokens per sentence.

Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 160–165
Copenhagen, Denmark, September 7, 2017.

2 Problem Definition

The NER is a classic sequence labeling problem, in which we are given a sentence, in the form of a sequence of tokens $\mathbf{w} = (w_1, w_2, ..., w_n)$, and we are required to output a sequence of token labels $\mathbf{y} = (y_1, y_2, ..., y_n)$. In this specific task, we use the standard BIO2 annotation, and each named entity chunk are classified into 6 categories, namely Person, Location (including GPE, facility), Corporation, Consumer good (tangible goods, or well-defined services), Creative work (song, movie, book, and so on) and Group (subsuming music band, sports team, and non-corporate organizations).

3 Approach

In this section, we will first introduce the overview of our proposed model and then present each part of the model in detail.

3.1 Overview

Figure 1 shows the overall structure of our proposed model, instead of solely using the original pretrained word embeddings as the final word representations, we construct a comprehensive word representation for each word in the input sentence. This comprehensive word representations contain the character-level sub-word information, the original pretrained word embeddings and multiple syntactical features. Then, we feed them into a Bidirectional LSTM layer, and thus we have a hidden state for each word. The hidden states are considered as the feature vectors of the words by the final CRF layer, from which we can decode the final predicted tag sequence for the input sentence.

3.2 Comprehensive Word Representations

In this subsection, we present our proposed comprehensive word representations. We first build character-level word representations from the embeddings of every character in each word using a bidirectional LSTM. Then we further incorporate the final word representation with the embedding of the syntactical information of each token, such as the part-of-speech tag, the dependency role, the word position in the sentence and the head position. Finally, we combine the original word embeddings with the above two parts to obtain the final comprehensive word representations.

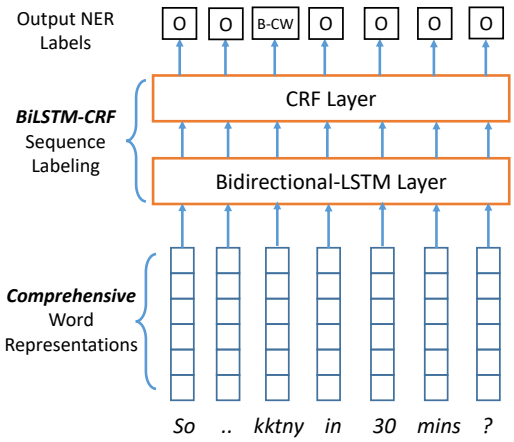

Figure 1: Overview of our approach.

3.2.1 Character-level Word Representations

In noisy user-generated text analysis, sub-word (character-level) information is much more important than that in normal text analysis for two main reasons: 1) People are more likely to use novel abbreviations and morphs to mention entities, which are often out of vocabulary and only occur a few times. Thus, solely using the original word-level word embedding as features to represent words is not adequate to capture the characteristics of such mentions. 2) Another reason why we have to pay more attention to character-level word representation for noisy text is that it is can capture the orthographic or morphological information of both formal words and Internet slang.

There are two main network structures to make use of character embeddings: one is CNN (Ma and Hovy, 2016) and the other is BiLSTM(Lample et al., 2016). BiLSTM turns to be better in our experiment on development dataset. Thus, we follow Lample et al. (2016) to build a BiLSTM network to encode the characters in each token as Figure 2 shows. We finally concatenate the forward embedding and backward embedding to the final character-level word representation.

3.2.2 Syntactical Word Representations

We argue that the syntactical information, such as POS tags and dependency roles, should also be explicitly considered as contextual features of each token in the sentence.

TweetNLP and TweeboParser (Owoputi et al., 2013; Kong et al., 2014) are two popular software to generate such syntactical tags for each token given a tweet. Given the nature of the noisy tweet text, a new set of POS tags and dependency

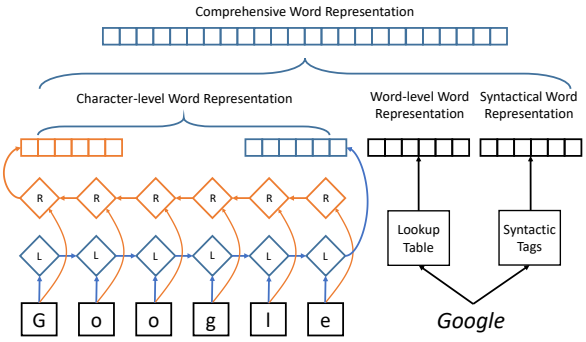

Figure 2: Illustration of comprehensive word representations.

trees are used in the tool, called Tweebank (Gimpel et al., 2011). See Table 1 for an example POS tagging. Since a tweet often contains more than one utterance, the output of TweeboParser will often be a multi-rooted graph over the tweet.

Word position embedding are included as well as it is widely used in other similar tasks, like relation classification (Xu et al., 2016). Also, head position embeddings are taken into account while calculating these embedding vectors to further enrich the dependency information. It tries to exclude these tokens from the parse tree, resulting a head index of -1.

After calculating all 4 types of embedding vectors (POS tags, dependency roles, word positions, head positions) for every tokens, we concatenate them to form a syntactical word representation.

Token	so	..	kktny	in	30	mins	?
POS	R	,	N	P	$	N	,
Position	1	2	3	4	5	6	7
Head	0	-1	0	3	6	4	-1

Table 1: Example of POS tagging for tweets.

3.2.3 Combination with Word-level Word Representations

After obtaining the above two additional word representations, we combine them with the original word-level word representations, which are just traditional word embeddings.

To sum up, our comprehensive word representations are the concatenation of three parts: 1) character-level word representations, 2) syntactical word representation and 3) original pretrained word embeddings.

3.3 BiLSTM Layer

LSTM based networks are proven to be effective in sequence labeling problem for they have access to both past and the future contexts. Whereas, hidden states in unidirectional LSTMs only takes information from the past, which may be adequate to classify the sentiment is a shortcoming for labeling each token. Bidirectional LSTMs enable the hidden states to capture both historical and future context information and then to label a token.

Mathematically, the input of this BiLSTM layer is a sequence of comprehensive word representations (vectors) for the tokens of the input sentence, denoted as $(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$. The output of this BiLSTM layer is a sequence of the hidden states for each input word vectors, denoted as $(\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n})$. Each final hidden state is the concatenation of the forward $\overleftarrow{\mathbf{h_i}}$ and backward $\overrightarrow{\mathbf{h_i}}$ hidden states. We know that

$$\overleftarrow{\mathbf{h_i}} = \text{lstm}(\mathbf{x_i}, \overleftarrow{\mathbf{h_{i-1}}}) \ , \ \overrightarrow{\mathbf{h_i}} = \text{lstm}(\mathbf{x_i}, \overrightarrow{\mathbf{h_{i+1}}})$$

$$\mathbf{h_i} = \left[\overleftarrow{\mathbf{h_i}} \ ; \ \overrightarrow{\mathbf{h_i}} \right]$$

3.4 CRF Layer

It is almost always beneficial to consider the correlations between the current label and neighboring labels since there are many syntactical constrains in natural language sentences. For example, I-PERSON will never follow a B-GROUP. If we simply feed the above mentioned hidden states independently to a Softmax layer to predict the labels, then such constrains will not be more likely to be broken. Linear-chain Conditional Random Field is the most popular way to control the structure prediction and its basic idea is to use a series of potential function to approximate the conditional probability of the output label sequence given the input word sequence.

Formally, we take the above sequence of hidden states $\mathbf{h} = (\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n})$ as our input to the CRF layer, and its output is our final prediction label sequence $\mathbf{y} = (y_1, y_2, ..., y_n)$, where y_i is in the set of all possible labels. We denote $\mathcal{Y}(\mathbf{h})$ as the set of all possible label sequences. Then we derive the conditional probability of the output sequence given the input hidden state sequence is

$$p(\mathbf{y}|\mathbf{h}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^{n} \exp(\mathbf{W}^T_{y_{i-1}, y_i} \mathbf{h} + \mathbf{b}_{y_{i-1}, y_i})}{\sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{h})} \prod_{i=1}^{n} \exp(\mathbf{W}^T_{y'_{i-1}, y'_i} \mathbf{h} + \mathbf{b}_{y'_{i-1}, y'_i})}$$

, where $\mathbf{W}$ and $\mathbf{b}$ are the two weight matrices and the subscription indicates that we extract the weight vector for the given label pair (y_{i-1}, y_i).

To train the CRF layer, we use the classic maximum conditional likelihood estimation to train our model. The final log-likelihood with respect to the weight matrices is

$$L(\mathbf{W}, \mathbf{b}) = \sum_{(\mathbf{h_i}, \mathbf{y_i})} \log p(\mathbf{y_i}|\mathbf{h_i}; \mathbf{W}, \mathbf{b})$$

Finally, we adopt the Viterbi algorithm for training the CRF layer and the decoding the optimal output sequence $\mathbf{y}^*$.

4 Experiments

In this section, we discuss the implementation details of our system such as hyper parameter tuning and the initialization of our model parameters. [2]

4.1 Parameter Initialization

For word-level word representation (i.e. the lookup table), we utilize the pretrained word embeddings[3] from GloVe(Pennington et al., 2014). For all out-of-vocabulary words, we assign their embeddings by randomly sampling from range $\left[-\sqrt{\frac{3}{\text{dim}}}, +\sqrt{\frac{3}{\text{dim}}}\right]$, where *dim* is the dimension of word embeddings, suggested by He et al.(2015). The random initialization of character embeddings are in the same way. We randomly initialize the weight matrices $\mathbf{W}$ and $\mathbf{b}$ with uniform samples from $\left[-\sqrt{\frac{6}{r+c}}, +\sqrt{\frac{6}{r+c}}\right]$, r and c are the number of the rows and columns, following Glorot and Bengio(2010). The weight matrices in LSTM are initialized in the same work while all LSTM hidden states are initialized to be zero except for the bias for the forget gate is initialized to be 1.0 , following Jozefowicz et al.(2015).

4.2 Hyper Parameter Tuning

We tuned the dimension of word-level embeddings from {50, **100**, 200}, character embeddings from {10, **25**, 50}, character BiLSTM hidden states (i.e. the character level word representation) from {20, **50**, 100}. We finally choose the bold ones. The dimension of part-of-speech tags, dependecny roles, word positions and head positions are all 5.

As for learning method, we compare the traditional SGD and Adam (Kingma and Ba, 2014). We found that Adam performs always better than SGD, and we tune the learning rate form {1e-2,**1e-3**,1e-4}.

4.3 Results

To evaluate the effectiveness of each feature in our model, we do the feature ablation experiments and the results are shown in Table 2.

Features	F1 (entity)	F1 (surface form)
Word	37.16	34.15
Char(LSTM)+Word	38.24	37.21
POS+Char(LSTM)+Word	40.01	37.57
Syntactical+Char(CNN)+Word	40.12	37.52
Syntactical+Char(LSTM)+Word	**40.42**	**37.62**

Table 2: Feature Ablation

In comparison with other participants, the results are shown in Table 3.

Team	F1 (entity)	F1 (surface form)
Drexel-CCI	26.30	25.26
MIC-CIS	37.06	34.25
FLYTXT	38.35	36.31
Arcada	39.98	37.77
Ours	**40.42**	**37.62**
SpinningBytes	40.78	39.33
UH-RiTUAL	**41.86**	**40.24**

Table 3: Result comparison

5 Related Work

Conditional random field (CRF) is a most effective approaches (Lafferty et al., 2001; McCallum and Li, 2003) for NER and other sequence labeling tasks and it achieved the state-of-the-art performance previously in Twitter NER (Baldwin et al., 2015). Whereas, it often needs lots of hand-craft features. More recently, Huang et al. (2015) introduced a similar but more complex model based on BiLSTM, which also considers hand-crafted features. Lample et al. (2016) further introduced using BiLSTM to incorporate character-level word representation. Whereas, Ma and Hovy (2016) replace the BiLSTM to CNN to build the character-level word representation. Limsopatham and Collier (2016), used similar model and achieved the best performance in the last shared task (Strauss et al., 2016). Based on the previous work, our system take more syntactical information into account, such as part-of-speech tags, dependency roles, token positions and head positions, which are proven to be effective.

[2] The detailed description of the evaluation metric and the dataset are shown in http://noisy-text.github.io/2017/emerging-rare-entities.html

[3] http://nlp.stanford.edu/data/glove.twitter.27B.zip

6 Conclusion

In this paper, we present a novel multi-channel BiLSTM-CRF model for emerging named entity recognition in social media messages. We find that BiLST-CRF architecture with our proposed comprehensive word representations built from multiple information are effective to overcome the noisy and short nature of social media messages.

References

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language* 44:61–83.

Timothy Baldwin, Young-Bum Kim, Marie Catherine De Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. *ACL-IJCNLP* 126:2015.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management* 51(2):32–49.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 42–47.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*. pages 249–256.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. pages 1026–1034.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991.

Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. pages 2342–2350.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A Smith. 2014. A dependency parser for tweets .

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 (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*. pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. pages 260–270.

Nut Limsopatham and Nigel Collier. 2016. Bidirectional lstm for named entity recognition in twitter messages.

Xuezhe Ma and Eduard H. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. pages 188–191. http://aclweb.org/anthology/W/W03/W03-0430.pdf.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Benjamin Strauss, Bethany E. Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv preprint arXiv:1601.03651* .

Boliang Zhang, Hongzhao Huang, Xiaoman Pan, Sujian Li, Chin-Yew Lin, Heng Ji, Kevin Knight, Zhen Wen, Yizhou Sun, Jiawei Han, and Bülent Yener. 2015. Context-aware entity morph decoding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 586–595.