Bilingual Character Representation for Efficiently Addressing Out-of-Vocabulary Words in Code-Switching Named Entity Recognition

Genta Indra Winata, Chien-Sheng Wu, Andrea Madotto, Pascale Fung
Center for Artificial Intelligence Research (CAiRE)
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{giwinata, cwuak, eeandreamad}@ust.hk, pascale@ece.ust.hk

Abstract

We propose an LSTM-based model with hierarchical architecture on named entity recognition from code-switching Twitter data. Our model uses bilingual character representation and transfer learning to address out-of-vocabulary words. In order to mitigate data noise, we propose to use token replacement and normalization. In the 3rd Workshop on Computational Approaches to Linguistic Code-Switching Shared Task, we achieved second place with 62.76% harmonic mean F1-score for English-Spanish language pair without using any gazetteer and knowledge-based information.

1 Introduction

Named Entity Recognition (NER) predicts which word tokens refer to location, people, organization, time, and other entities from a word sequence. Deep neural network models have successfully achieved the state-of-the-art performance in NER tasks (Cohen; Chiu and Nichols, 2016; Lample et al., 2016; Shen et al., 2017) using monolingual corpus. However, learning from code-switching tweets data is very challenging due to several reasons: (1) words may have different semantics in different context and language, for instance, the word "cola" can be associated with product or "queue" in Spanish (2) data from social media are noisy, with many inconsistencies such as spelling mistakes, repetitions, and informalities which eventually points to Out-of-Vocabulary (OOV) words issue (3) entities may appear in different language other than the matrix language. For example "todos los Domingos en Westland Mall" where "Westland Mall" is an English named entity.

Our contributions are two-fold: (1) bilingual character bidirectional RNN is used to capture character-level information and tackle OOV words issue (2) we apply transfer learning from monolingual pre-trained word vectors to adapt the model with different domains in a bilingual setting. In our model, we use LSTM to capture long-range dependencies of the word sequence and character sequence in bilingual character RNN. In our experiments, we show the efficiency of our model in handling OOV words and bilingual word context.

2 Related Work

Convolutional Neural Network (CNN) was used in NER task as word decoder by Collobert et al. (2011) and a few years later, Huang et al. (2015) introduced Bidirectional Long-Short Term Memory (BiLSTM) (Sundermeyer et al., 2012). Character-level features were explored by using neural architecture and replaced hand-crafted features (Dyer et al., 2015; Lample et al., 2016; Chiu and Nichols, 2016; Limsopatham and Collier, 2016). Lample et al. (2016) also showed Conditional Random Field (CRF) (Lafferty et al., 2001) decoders to improve the results and used Stack memory-based LSTMs for their work in sequence chunking. Aguilar et al. (2017) proposed multi-task learning by combining Part-of-Speech tagging task with NER and using gazetteers to provide language-specific knowledge. Character-level embeddings were used to handle the OOV words problem in NLP tasks such as NER (Lample et al., 2016), POS tagging, and language modeling (Ling et al., 2015).

3 Methodology

3.1 Dataset

For our experiment, we use English-Spanish (ENG-SPA) Tweets data from Twitter provided by

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 110–114
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

Table 1: OOV words rates on ENG-SPA dataset before and after preprocessing

	Train		Dev		Test
	All	Entity	All	Entity	All
Corpus	-	-	18.91%	31.84%	49.39%
FastText (eng) (Mikolov et al., 2018)	62.62%	16.76%	19.12%	3.91%	54.59%
+ FastText (spa) (Grave et al., 2018)	49.76%	12.38%	11.98%	3.91%	39.45%
+ token replacement	12.43%	12.35%	7.18%	3.91%	9.60%
+ token normalization	**7.94%**	**8.38%**	**5.01%**	**1.67%**	**6.08%**

Aguilar et al. (2018). There are nine different named-entity labels. The labels use IOB format (Inside, Outside, Beginning) where every token is labeled as `B-label` in the beginning and follows with `I-label` if it is inside a named entity, or `O` otherwise. For example "Kendrick Lamar" is represented as `B-PER I-PER`. Table 2 and Table 3 show the statistics of the dataset.

Table 2: Data Statistics for ENG-SPA Tweets

	Train	Dev	Test
# Words	616,069	9,583	183,011

Table 3: Entity Statistics for ENG-SPA Tweets

Entities	Train	Dev
# Person	4701	75
# Location	2810	10
# Product	1369	16
# Title	824	22
# Organization	811	9
# Group	718	4
# Time	577	6
# Event	232	4
# Other	324	6

"Person", "Location", and "Product" are the most frequent entities in the dataset, and the least common ones are "Time", "Event", and "Other" categories. 'Other" category is the least trivial among all because it is not well clustered like others.

3.2 Feature Representation

In this section, we describe word-level and character-level features used in our model.

Word Representation: Words are encoded into continuous representation. The vocabulary is built from training data. The Twitter data are very noisy, there are many spelling mistakes, irregular ways to use a word and repeating characters. We apply several strategies to overcome the issue. We use 300-dimensional English (Mikolov et al., 2018) and Spanish (Grave et al., 2018) FastText pre-trained word vectors which comprise two million words vocabulary each and they are trained using Common Crawl and Wikipedia. To create the shared vocabulary, we concatenate English and Spanish word vectors.

For preprocessing, we propose the following steps:

1. **Token replacement:** Replace user hashtags (#user) and mentions (@user) with "USR", and URL (https://domain.com) with "URL".

2. **Token normalization:** Concatenate Spanish and English FastText word vector vocabulary. Normalize OOV words by using one out of these heuristics and check if the word exists in the vocabulary sequentially

   (a) Capitalize the first character

   (b) Lowercase the word

   (c) Step (b) and remove repeating characters, such as *"hellooooo"* into *"hello"* or *"lololol"* into *"lol"*

   (d) Step (a) and (c) altogether

Then, the effectiveness of the preprocessing and transfer learning in handling OOV words are analyzed. The statistics is showed in Table 1. It is clear that using FastText word vectors reduce the OOV words rate especially when we concatenate the vocabulary of both languages. Furthermore, the preprocessing strategies dramatically decrease the number of unknown words.

Character Representation: We concatenate all possible characters for English and Spanish, including numbers and special characters. English and Spanish have most of the characters in common, but, with some additional unique Spanish characters. All cases are kept as they are.

3.3 Model Description

In this section, we describe our model architecture and hyper-parameters setting.

Bilingual Char-RNN: This is one of the approaches to learn character-level embeddings without needing of any lexical hand-crafted features. We use an RNN for representing the word with character-level information (Lampe et al., 2016). Figure 1 shows the model architecture. The inputs are characters extracted from a word and every character is embedded with d dimension vector. Then, we use it as the input for a Bidirectional LSTM as character encoder, wherein every time step, a character is input to the network. Consider a_t as the hidden states for word t.

$$a_t = (a_1^1, a_t^2, ..., a_t^{\mathrm{V}})$$

where V is the character length. The representation of the word is obtained by taking a_t^{V} which is the last hidden state.

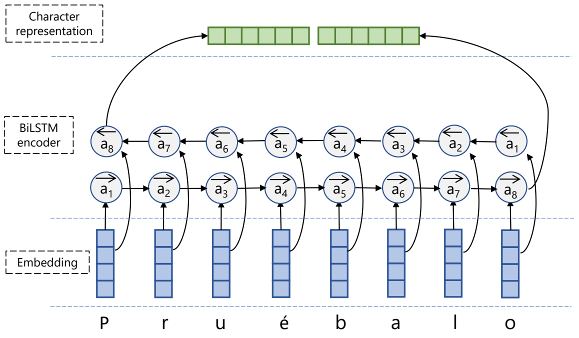

Figure 1: Bilingual Char-RNN architecture

Main Architecture: Figure 2 presents the overall architecture of the system. The input layers receive word and character-level representations from English and Spanish pre-trained FastText word vectors and Bilingual Char-RNN. Consider **X** as the input sequence:

$$\mathbf{X} = (x_1, x_2, ..., x_{\mathrm{N}})$$

where N is the length of the sequence. We fix the word embedding parameters. Then, we concatenate both vectors to get a richer word representation u_t. Afterwards, we pass the vectors to bidirectional LSTM.

$$u_t = x_t \oplus a_t$$

$$\overrightarrow{h_t} = \overrightarrow{\mathrm{LSTM}}(u_t, \overrightarrow{h_{t-1}}), \overleftarrow{h_t} = \overleftarrow{\mathrm{LSTM}}(u_t, \overleftarrow{h_{t-1}})$$

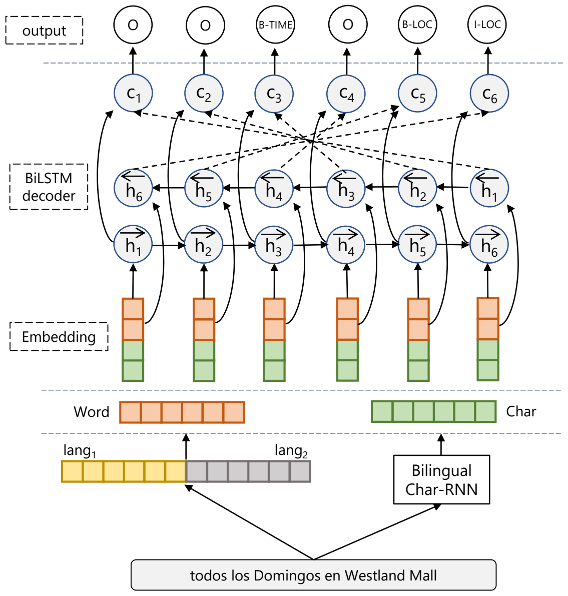

Figure 2: Main architecture

$$c_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}$$

where $\oplus$ denotes the concatenation operator. Dropout is applied to the recurrent layer. At each time step we make a prediction for the entity of the current token. A softmax function is used to calculate the probability distribution of all possible named-entity tags.

$$y_t = \frac{e^{c_t}}{\sum_{j=1}^{T} e^{c_j}}, \text{ where } j = 1, .., \mathrm{T}$$

where y_t is the probability distribution of tags at word t and T is the maximum time step. Since there is a variable number of sequence length, we padded the sequence and applied mask when calculating cross-entropy loss function. Our model does not use any gazetteer and knowledge-based information, and it can be easily adapted to another language pair.

3.4 Post-processing

We found an issue during the prediction where some words are labeled with `O`, in between `B-label` and `I-label` tags. Our solution is to insert `I-label` tag if the tag is surrounded by `B-label` and `I-label` tags with the same entity category. Another problem we found that many `I-label` tags are paired with `B-label` in different categories. So, we replace `B-label` category tag with corresponding `I-label` category tag. This step improves the result of the pre-

Table 4: Results on ENG-SPA Dataset (‡ result(s) from the shared task organizer (Aguilar et al., 2018) † without token normalization)

Model	Features	F1 Dev	F1 Test
Baseline‡	Word	-	53.2802%
BiLSTM†	Word + Char-RNN	46.9643%	53.4759%
BiLSTM	FastText (eng)	57.7174%	59.9098%
BiLSTM	FastText (eng-spa)	57.4177%	60.2426%
BiLSTM	+ Char-RNN	65.2217%	61.9621%
+ post		**65.3865%**	**62.7608%**
Competitors‡			
IIT BHU (1st place)	-	-	63.7628% (+1.0020%)
FAIR (3rd place)	-	-	62.6671% (- 0.0937%)

diction on the development set. Figure 3 shows the examples.

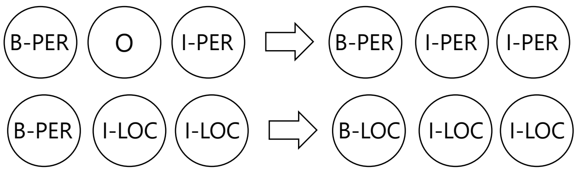

Figure 3: Post-processing examples

3.5 Experimental Setup

We trained our LSTM models with a hidden size of 200. We used batch size equals to 64. The sentences were sorted by length in descending order. Our embedding size is 300 for word and 150 for characters. Dropout (Srivastava et al., 2014) of 0.4 was applied to all LSTMs. Adam Optimizer was chosen with an initial learning rate of 0.01. We applied time-based decay of $\sqrt{2}$ decay rate and stop after two consecutive epochs without improvement. We tuned our model with the development set and evaluated our best model with the test set using harmonic mean F1-score metric with the script provided by Aguilar et al. (2018).

4 Results

Table 4 shows the results for ENG-SPA tweets. Adding pre-trained word vectors and character-level features improved the performance. Interestingly, our initial attempts at adding character-level features did not improve the overall performance, until we apply dropout to the Char-RNN. The performance of the model improves significantly after transfer learning with FastText word vectors while it also reduces the number of OOV words in the development and test set. The margin between ours and first place model is small, approximately 1%.

We try to use sub-words representation from Spanish FastText (Grave et al., 2018), however, it does not improve the result since the OOV words consist of many special characters, for example, *"/IAtrevido/Provocativo"*, *"Twets/wek"*, and possibly create noisy vectors and most of them are not entity words.

5 Conclusion

This paper presents a bidirectional LSTM-based model with hierarchical architecture using bilingual character RNN to address the OOV words issue. Moreover, token replacement, token normalization, and transfer learning reduce OOV words rate even further and significantly improves the performance. The model achieved 62.76% F1-score for English-Spanish language pair without using any gazetteer and knowledge-based information.

Acknowledgments

This work is partially funded by ITS/319/16FP of the Innovation Technology Commission, HKUST 16214415 & 16248016 of Hong Kong Research Grants Council, and RDC 1718050-0 of EMOS.AI.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Overview of the CALCS 2018 Shared Task: Named

Entity Recognition on Code-switched Data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, Melbourne, Australia. Association for Computational Linguistics.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

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

Zhilin Yang Ruslan Salakhutdinov William Cohen. Multi-task cross-lingual sequence tagging from scratch.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 334–343.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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

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 NAACL-HLT*, pages 260–270.

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

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.