IIT (BHU) Submission for the ACL Shared Task on Named Entity Recognition on Code-switched Data

Shashwat Trivedi[*]**, Harsh Rangwani**[*]**and Anil Kumar Singh**

Indian Institute of Technology (Banaras Hindu University), Varanasi, India
{shashwat.trivedi.cse15, harsh.rangwani.cse15, aksingh.cse}@iitbhu.ac.in

Abstract

This paper describes the best performing system for the shared task on Named Entity Recognition (NER) on code-switched data for the language pair Spanish-English (ENG-SPA). We introduce a gated neural architecture for the NER task. Our final model achieves an F1 score of 63.76%, outperforming the baseline by 10%.

1 Introduction

Named Entity Recognition (NER) is an important Natural Language Processing task, which involves extracting named entities (i.e., Names of Persons, Entities, Organizations etc.) from the provided text, and the classification of entities into a certain number of predefined categories. The extracted entities provide us with the important information about the content of the text (Nadeau and Sekine, 2007). For example, "New Delhi is famous for its historical past.". The extracted entity (*New Delhi*) gives us an idea that the text is associated with the *location* called *New Delhi*. The ability of NER to extract this useful information makes it an essential part of the Information Extraction pipeline.

The social media platforms like Twitter, Reddit etc. have become a massive source of information due to their growth in the recent past. Performing NER on social texts can be challenging due to the unstructured and colloquial nature of social texts. Various attempts have been made in the past to solve the problem of NER on social texts (Derczynski et al., 2017; Strauss et al., 2016). However, most of the previous systems were developed to work with monolingual texts (Ritter et al., 2011; Lin et al., 2017), ignoring the phenomena of code-switching (i.e., switching between different languages within a sentence), which is quite prevalent in social media texts.

This paper describes our system for Named Entity Recognition Shared Task on English-Spanish Code-switched tweets held at the ACL 2018 Workshop on Computational Approaches to Linguistic Code-switching. The task involves categorizing a token into 19 different categories. More details about the task can be found in the task description paper (Aguilar et al., 2018).

We use a novel architecture based on **gating** of character-based representations and word-based representations of a token (Yang et al., 2016). The character-based representation is generated using a 'Char CNN' (Zhang et al., 2015) and the word-based representation is generated using an LSTM (Hochreiter and Schmidhuber, 1997). Furthermore, the activations from the last but one layer of the neural networks, trained with different hyper-parameters, are ensembled and then are passed as features to a Conditional Random Field (CRF) classifier for final predictions. We make use of English Twitter embeddings (Godin et al., 2015), aligned with the Spanish embeddings (Bojanowski et al., 2016) as described in Section 2.1.

Our final submitted system achieves the best result on the shared task with 63.76% F1-score.

2 Proposed Approach

This section describes feature representations, model description and the ensembling technique in detail.

2.1 Feature Representation

The following representations are used to capture overall information for each token: Word, Character and Lexical representations.

Word Representation: Word representations are created using concatenation of two separate representations, one based on the pre-trained word vec-

[*] These authors have equal contribution to the paper

Proceedings of The Third Workshop on Computational Approaches to Code-Switching, pages 148–153
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

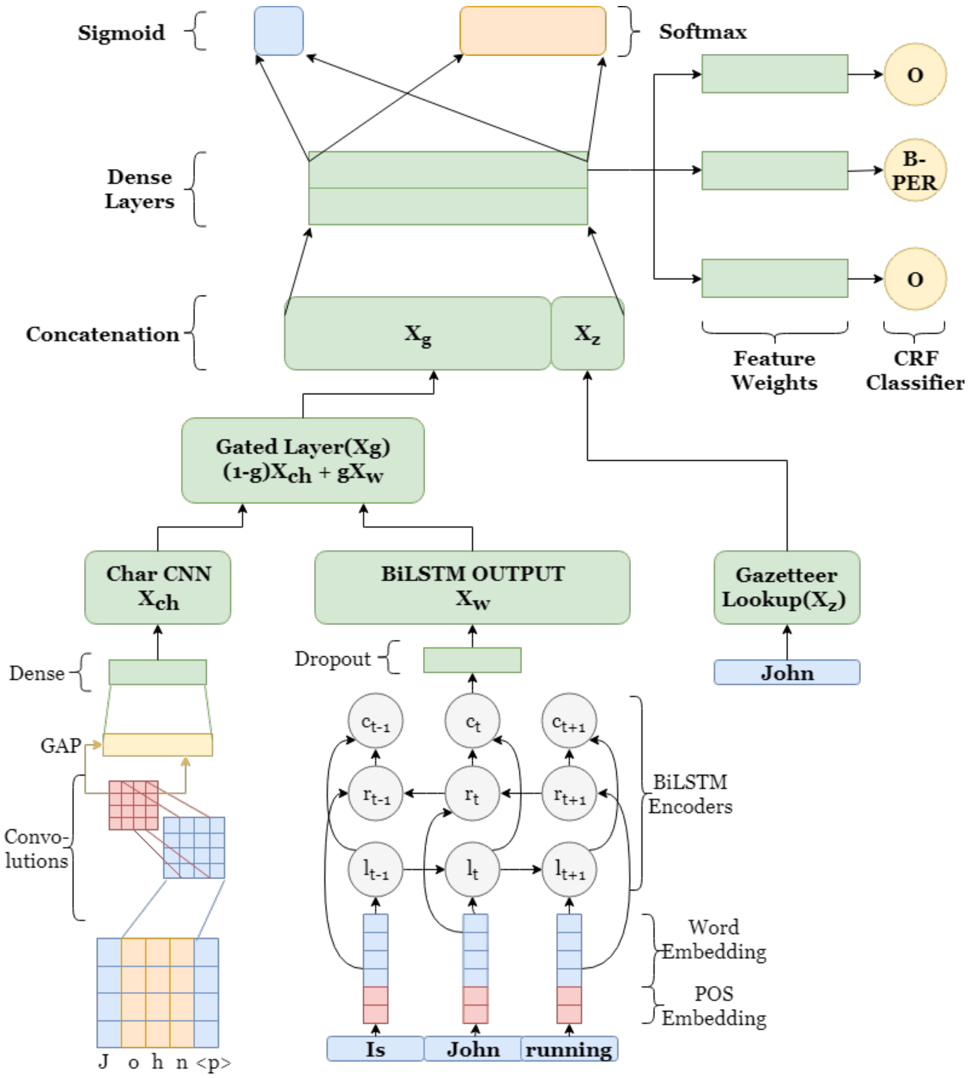

Figure 1: Final Architecture Of The System

tors and the other based on Part-of-Speech (POS) tag embeddings.

For the word vector representation, we use Spanish FastText word vectors (Bojanowski et al., 2016) of 300-dimensions, trained on Wikipedia and pre-trained word embeddings (Godin et al., 2015) of 400-dimensions, trained on 400 million tweets. We use a Principal Component Analysis (PCA) based algorithm suggested by Raunak (2017) to reduce the dimensions of the Twitter word vectors. Since these word vectors are in different vector spaces, we use Singular Value Decomposition (SVD) (Smith et al., 2017) for aligning these two embeddings to represent them in a single vector space.
For POS tagging, we use the CMU Part-of-Speech tagger (Owoputi et al., 2013). Each POS tag is represented as a vector of dimension *dim*. The vectors corresponding to the POS tags are initialized randomly with uniform distribution range $\left[-\sqrt{3/dim}, \sqrt{3/dim}\right]$ as suggested by He et al. (2015). The word vector corresponding to the token is concatenated with the vector corresponding to the POS tag of the token to obtain the final vector representation.

For obtaining the label for each token, we provide a composite vector as an input to the model. The composite vector is generated by concatenation of word representations of adjacent tokens (one on each side) with its own, same as a trigram.
Character Representation: At the character level, we represent each token as a sequence of character embeddings. These embeddings are initialized randomly with uniform distribution range, similar to POS tag embeddings. In the model, they are kept *trainable* to learn the representation cor-

responding to each character. Each token is either truncated or post-padded to generate a token of 20 characters.

Lexical Representation: We use the gazetteer provided by Mishra and Diesner (2016) and some Spanish gazetteers of our own to provide world knowledge to our model. Top 1000 celebrity Twitter handles from this list[1] are also added. We represent gazetteer input for a token as a 19 dimensional vector, one binary value corresponding to each class. The binary bit represents the presence (1) or absence (0) of the token in the gazetteer (i.e. word list) of the respective class.

2.2 Model Description

BiLSTM for Word Representation: We use Bidirectional LSTM (Dyer et al., 2015) in the model to learn the contextual relationship between the words. Word representations described earlier are used as input to this layer. The BiLSTM layer consists of two LSTM layers having 3 units each. With one layer connected in the forward direction and the other layer connected in the backward direction, this captures the information from the past and the future (Ma and Hovy, 2016). The outputs of both forward and backward LSTM are then concatenated to produce a final single embedding for the input token. We vary recurrent dropouts (Gal and Ghahramani, 2016), input dropouts and output dropouts as shown in the Table 1, across three different models. The gate layer is fed with the output of this layer (X$_w$).

Convolution Network for Character Representation: We use a CNN-architecture to learn the character based representation of a word. The character embeddings of a token, denoted as $\mathbb{R}^{d \times l}$, where *d* is the dimension of a single character's embedding and *l* is the max length of the token, is fed to a 2-stacked convolutional layer, both activated using ReLU function. Its results are then pushed into a pooling layer. We applied two different pooling techniques, specified in the Table 1, across different models. The output of the pooling layer serves as an input to a dense layer, whose activation function (*Char dense layer activation*) is varied as shown in Table 1. Finally, we use the output of the dense layer (X_{ch}) as an input to the gate layer.

Gate Layer: The concatenation of word representations and POS tag embeddings is used as input to a *sigmoid* dense layer. The value of the sigmoid output controls the relative contribution of the character and word representation in the final representation of the token. Following the work of Miyamoto and Cho (2016), the output of this layer g is used to take the weighted average of Bi-LSTM network output (X_g) and the convolutional network output (X_{ch}):

$$g = \sigma(v_g^T X_g + b_g)$$

$$X = (1 - g)X_{ch} + gX_g$$

where v_g is the trainable weight vector, b_g is the bias and $\sigma(\cdot)$ is the sigmoid function. The result of this layer X is then concatenated with the gazetteer embeddings of the token.

Fully Connected Network: We use two fully connected networks after the concatenation of the gate network output and gazetteer embeddings. The number of dense units is kept fixed to 100 each. The activation function is varied according to Table 1 for producing different models.

Multitask Learning: Multitask learning has been shown as a good way to regularize models (Baxter, 2000; Collobert and Weston, 2008). Following the work of Aguilar et al. (2017), we split the task into Named Entity (NE) categorization (classifying a token into one of the NE classes) and NE segmentation (classifying token as NE or Not-NE). We passed the dense layer's output as input to these final classification layers. A softmax layer with 19 classes is used for the categorization task and a single sigmoid neuron is used for the segmentation task as depicted in Figure 1. The cross-entropy losses for these tasks are added to yield total loss for the model.

2.3 Conditional Random Fields and Ensembling

Linear-chain CRF classifier takes advantage of the sequence information to tag a token with the most probable label (Lafferty et al., 2001). Following Aguilar et al. (2017), we use the activations of second common dense layer as input feature vector for the CRF classifier. The CRF classifier produces better results than the normal softmax classification and also reduces the number of invalid predictions (i.e., I-PER tag without a B-PER tag). For preparing the model ensemble, we make use of

[1] https://gist.github.com/mbejda/9c3353780270e7298763

Table 1: Hyper-parameters for the Models and Ensemble Results

Hyper-Parameters	Model-1	Model-2	Model-3
POS and character embeddings dropout	0.500	0.500	0.247
POS embeddings dimension	50	128	128
Character embeddings dimension	100	128	128
Pooling layer	$^*GAP$	GAP	^+GMP
Char dense layer activation	ReLU	ReLU	tanh
Recurrent dropouts	0.500	0.500	0.823
BiLSTM input dropout	-	-	0.0654
BiLSTM output dropout	0.500	0.500	0.018
Dense layer activation	ReLU	ReLU	tanh
Preprocessing of Test-data	X	Y	Y
Optimiser	$^\#$nadam	nadam	rmsprop
Results (F1 score)	**61.18%**	**61.89%**	**60.23%**
Overall Ensemble of Model1 + Model2 + Model3 (F1 Score)		**63.76%**	

$^*GAP$:Global Average Pooling ^+GMP:Global Max Pooling
$^\#$nadam is adam rmsprop with nesterov momentum (Dozat, 2016)

unweighted averaging of the activations generated by the networks described in Table 1.

2.4 Experimental Settings

2.4.1 Pre-processing

The data is pre-processed by doing the following replacements: All URLs are replaced with $\langle url\rangle$. All hashtags are replaced with $\langle hashtag\rangle$. Digits are replaced with the $\langle number\rangle$ token. Apostrophes are removed. Finally, emoticons are replaced with their respective meaning, for example, ‘:-)’ with $\langle smile\rangle$.

2.4.2 Hyper-parameters

Different hyper-parameters are used to produce different models for ensembling. We set the following parameters as the same across all the models: 64 filters, kernel size of 3 and ReLU activation in convolutional network (Section 2.2), along with 50 hidden units in the BiLSTM network (Section 2.2).

Other hyperparameters are set according to the Table 1 for the respective models. All models are trained for 15 epochs with a batch size of 512. The CRF classifier is used with the following parameters: L1 penalty: 1.0, L2 penalty: 1e-3 for 80 epochs.

Hyper-parameters for Model-3 are obtained by a random search using hyperas[2]. Hyper-parameters for the other two models are set based on our own experimental observations. All our models are implemented using the Deep Learning library Keras[3].

3 Results and Discussion

We compare our final results with the RNN baseline, which is the official baseline of the task (Aguilar et al., 2018). The major highlights of our results are described below.

Table 2: Results in Different Categories

Models Used	Precision	Recall	F1
Event	37.50%	13.33%	19.67%
Group	38.36%	28.87%	32.94%
Location	70.31%	72.45%	71.37%
Organization	58.14%	24.75%	34.72%
Other	11.11%	1.72%	2.99%
Person	79.26%	77.87%	78.56%
Product	63.43%	44.16%	52.07%
Time	30.67%	30.46%	30.56%
Title	31.85%	19.46%	24.16%
Overall	**68.73%**	**59.47%**	**63.76%**
Baseline	**-**	**-**	**53.28%**

- Our model achieves an F1-score of 63.76%, which beats the baseline by around 10% on the test set. Our results depict the effectiveness of the use of gated neural architecture for Named Entity Recognition. Our system ranked first among the 8 systems submitted for the task.

[2] https://github.com/hyperopt/hyperopt

[3] https://github.com/keras-team/keras

- The system performance on the various class of entities is displayed in Table 2. Our model shows poor performance in Title, Other and Event categories. This may be attributed to both the diverse set of patterns present, and the unavailability of a large number of samples of these categories.

4 Conclusion

In this paper, we describe a gated neural network for performing NER on code-switched social media text. Our model involves the usage of SVD to align word representations of English and Spanish words. Furthermore, we also describe a novel way of ensembling activations of the last but one layer for achieving better results. Our model is described in full detail in this paper to ensure the replication of results. The final system performs the best among all the participating systems.
In future, we would like to experiment with various other ways of combining character and word representations (e.g. Fine Grained Gating (Zhang et al., 2015), Highway Networks (Liang et al., 2017) etc.) for the NER task.

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*. Association for Computational Linguistics, Melbourne, Australia.

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.

Jonathan Baxter. 2000. A model of inductive bias learning. *J. Artif. Int. Res.* 12(1):149–198. http://dl.acm.org/citation.cfm?id=1622248.1622254.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. pages 140–147.

Timothy Dozat. 2016. Incorporating nesterov momentum into adam. In *Proceedings of ICLR 2016 Workshop*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075* .

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*. pages 1019–1027.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*. pages 146–153.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. pages 1026–1034.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

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

Dongyun Liang, Weiran Xu, and Yinge Zhao. 2017. Combining word-level and character-level representations for relation classification of informal text. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. pages 43–47.

Bill Y Lin, Frank Xu, Zhiyi Luo, and Kenny Zhu. 2017. Multi-channel bilstm-crf model for emerging named entity recognition in social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. pages 160–165.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR* abs/1603.01354. http://arxiv.org/abs/1603.01354.

Shubhanshu Mishra and Jana Diesner. 2016. Semi-supervised named entity recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. The COLING 2016 Organizing Committee, pages 203–212. http://aclweb.org/anthology/W16-3927.

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. *CoRR* abs/1606.01700. http://arxiv.org/abs/1606.01700.

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

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.

Vikas Raunak. 2017. Effective dimensionality reduction for word embeddings. *CoRR* abs/1708.03629. http://arxiv.org/abs/1708.03629.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 1524–1534.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR* abs/1702.03859. http://arxiv.org/abs/1702.03859.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the wnut16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. pages 138–144.

Z. Yang, B. Dhingra, Y. Yuan, J. Hu, W. W. Cohen, and R. Salakhutdinov. 2016. Words or characters? fine-grained gating for reading comprehension. *ArXiv e-prints* http://adsabs.harvard.edu/abs/2016arXiv161101724Y.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR* abs/1509.01626. http://arxiv.org/abs/1509.01626.