InferNER: an attentive model leveraging the sentence-level information for Named Entity Recognition in Microblogs

Moemmur Shahzad,[1] Ayesha Amin,[2] Diego Esteves,[3] Axel-Cyrille Ngonga Ngomo,[1]
[1]Universität Paderborn [2]Universität des Saarlandes [3]Farfetch
moemmur@mail.uni-paderborn.de, ayam00001@stud.uni-saarland.de, diegoesteves@gmail.com,
axel.ngonga@upb.de

Abstract

We investigate the problem of named entity recognition in the user-generated text such as social media posts. This task is rendered particularly difficult by the restricted length and limited grammatical coherence of this data type. Current state-of-the-art approaches rely on external sources such as gazetteers to alleviate some of these restrictions. We present a neural model able to outperform state of the art on this task without recurring to gazetteers or similar external sources of information. Our approach relies on word-, character-, and sentence-level information for NER in short-text. Social media posts like tweets often have associated images that may provide auxiliary context relevant to understand these texts. Hence, we also incorporate visual information and introduce an attention component which computes attention weight probabilities over textual and text-relevant visual contexts separately. Our model outperforms the current state of the art on various NER datasets. On WNUT 2016 and 2017, our model achieved 53.48% and 50.52% F1 score, respectively. With Multimodal model, our system also outperforms the current SOTA with an F1 score of 74% on the multimodal dataset. Our evaluation further suggests that our model also goes beyond the current state-of-the-art on newswire data, hence corroborating its suitability for various NER tasks.

Introduction

Named Entity Recognition (NER) is a fundamental preprocessing step in many natural language processing applications. The subtask of information extraction concerns identifying and classification named entities mentioned in the natural language text into predefined categories (e.g., Person, Location, product, and others.). Commonly, NER is useful in recommendation and search engines where it helps content classification based on the subject and theme of the web documents.

Microblogging platforms such as Twitter have gained tremendous attention in recent years due to the high volume of content made available through them (Derczynski et al. 2014). However, microblogs often do not abide by the syntactic and semantic conventions of the language they are written in (Derczynski et al. 2014; Peres, Esteves, and Maheshwari 2017; Esteves et al. 2018). Instead, they use diverse writing styles characterized by unorthodox capitalization, the frequent use of emotion icons and abbreviations, incomplete phrases, spelling/grammatical errors, and hashtags. Ambiguity is another challenge, for example in the microblog, "*manchester nailed it.. #UCL*" here word *manchester* is a city as well as a football league team. With concise context, it is often challenging to interpret these texts. A direct consequence of aforementioned characteristics of natural language text makes NER on microblogs a non-trivial task (Limsopatham and Collier 2016; Partalas et al. 2016; Aguilar et al. 2017). However, NER on user-generated content is crucial for other information extraction tasks such as relation extraction and applications such as question answering.

Many previous approaches to NER in user-generated text either rely on external resources or text normalization (see, e.g., (Aguilar et al. 2017; Partalas et al. 2016)). In contrast to the previous systems, InferNER does not rely on external sources or any prepossessing techniques. Hence, it can be easily trained and ported to datasets and domains other than the ones it was tested on. Most of the previous neural approaches employ conditional random field (CRF) for NER on Microblogs. Previous works claimed that transfer learning from network to CRF boosts the NER performance (See, e.g., (Aguilar et al. 2019)). However, we achieve similar or better results without the CRF classifier in most cases. The addition of CRF led to adverse outcomes in our case. Our approach not only considers word and character-level representations but also, sentence-level context. We provide the sentence-level context for each word in the input sequence. This approach helps to disambiguate words that might have many different meanings to the best of our knowledge. Furthermore, we also employ auxiliary context from images to better understand these microblog texts. We present an attention module that computes attention over heterogeneous modalities (text and images) in segregation to yield a summed vector to present state-of-the-art results. Our contributions can be summarized as follows:

- We present a network that leverages the sentence-level context.
- We combine textual information with visual clues using a neural architecture with segregated attention to outperform state of the art on current benchmark datasets for NER on short texts.
- Our ablation study elucidates the most important features

when extracting entities from short texts and can be used as a foundation for future works on NER for short texts.

Related Work

In recent years, many proposed systems employ neural architectures for NER in microblogs. Most architectures use bidirectional long short term memory (BiLSTM) with a CRF at the end. For instance, the work in (Limsopatham and Collier 2016; Aguilar et al. 2017; von Däniken and Cieliebak 2017; Moon, Neves, and Carvalho 2018; Zhang et al. 2018) employ BiLSTM-CRF architecture. Furthermore, many WNUT [1] submissions relied on a gazetteer as a supplementary source. Besides, many of the participants to these challenges applied some preprocessing (e.g., Text normalization) to minimize the linguistic complexity of the microblogs they were to process (see, e.g., (Partalas et al. 2016)). von Däniken and Cieliebak (2017) was first to employ sentence-level information for NER in short-text. They used sent2vec (Pagliardini, Gupta, and Jaggi 2017) to provide supplementary context. However, their reported results were lower than their base system. The reason could be the way these supplementary context has been incorporated. Aguilar et al. (2018; 2019) utilized phonetic or phonological representations of informal words to model noise and achieved state-of-the-art results on user-generated text. Other recent works employ pooled contextualized embeddings (Akbik, Bergmann, and Vollgraf 2019), adversarial discriminative model to address the low-resource NER problem (Zhou et al. 2019), and handling label mistakes with CrossWeigh framework (Wang et al. 2019) to present state-of-the-art results on different NER datasets. These recent approaches still built on traditional BiLSTM-CRF architecture and need additional resources such as data. Our comprehensive benchmark with top-performing baselines is presented in Table 1.

Methods

Feature Extraction

We utilize word, character and sentence embeddings for the NER task. Additionally, we consider auxiliary context such as sentence-level information and context from images.

2-Stacked BiLSTM Word Encoder. We obtain word representations from the 2-stacked BiLSTM layers using deep contextualized representations (ELMo) (Peters et al. 2018) set to 1024 dimensions. We obtain the word representations by taking the sum of outputs of two layers stacked together. To the best of our knowledge, such feature extractors better leverage ELMo embeddings.

Consider an input sequence $x_t = [x_1, x_2, ..., x_n]$ where n is the sentence maximum length s. We first obtain an embedding matrix $\epsilon(x_t)$ and is then passed to a Bidirectional LSTM as given below:

$$r_o = BLSTM(\epsilon(x_t))$$

The output r_o is then passed to another Bidirectional LSTM layer similar to the one defined before. The output of the second layer is given by,

$$r_o^{'} = BLSTM(r_o)$$

The final word representation is the addition of the output from the first RNN and the last RNN which is given by:

$$w_t = r_o \oplus r_o^{'}. \quad (1)$$

Here, $\oplus$ stands for the addition of tensors. w_t is the final word representation.

Character-Level Encoder. We extract character representations using a CNN. Since the data lacks implicit linguistic formalism, therefore, we utilize orthographic representations. First, we generate the orthographic sentence using set of rules as in (Limsopatham and Collier 2016; Aguilar et al. 2017). We then initialize the character sequences with a random uniform distribution of $\left[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}\right]$ to a lookup table to output a character embedding of 30 dimensions (Aguilar et al. 2017; Ma and Hovy 2016). Not all words have an equal number of characters. Therefore, we computed the maximum character length of the word and applied character padding. Secondly, following Aguilar et al. (2017) we applied 2-stacked convolutions over character embedding and used *global average pooling* operation similar to Zhou et al.; Aguilar et al. (2015; 2017) to capture information from the feature map. Lastly, for a given input, we first obtain the embedding embedding matrix ϵ_c and finally we obtain the character-level representations from the final fully-connected layer that has a Rectifier Linear Unit (ReLU) as an activation function:

$$c_t = ReLU(W_{\epsilon_c}\epsilon_c + b_{\epsilon_c}) \quad (2)$$

where ϵ_c and c_t represents the embedding matrix and output of the fully-connected layer, respectively.+

Sentence-level Encoder. In order to leverage more contextual information, we use sentence-level encoder to extract sentence representation. This helps our model to consider sentence-level context while predicting the named entity for the word token. Cer et al. (2018) proposed a new method for computing sentence vectors called *Universal Sentence Encoder (USE)*. Recent evaluations suggest that *USE* improve the performance of machine learning algorithms on several downstream tasks. We incorporate more semantic features in the form of sentence embeddings. For instance, we use 512-dimensional embedding model trained with a transformer encoder. For a given sentence input x_s, we first obtain an embedding matrix ϵ_s. We then employ a feed-forward Dense network with ReLU as the activation function to obtain sentence representation s_o as given as below:

$$s_o = ReLU(W_{\epsilon_s}\epsilon_s + b_{\epsilon_s}) \quad (3)$$

Image Features Image features carry information about what objects are depicted in the image, providing an additional context for understanding text. We use the InceptionV3 (Szegedy et al. 2015) model trained on the ImageNet dataset. Image features had the dimension of $8 \times 8 \times 2048$, where 2048 is the dimension of a vector at each region N and 8×8 are the number of regions in the output image vector. The images were resized to 299×299 and extract features from the final convolution layer of InceptionV3.

[1] http://noisy-text.github.io/2020/

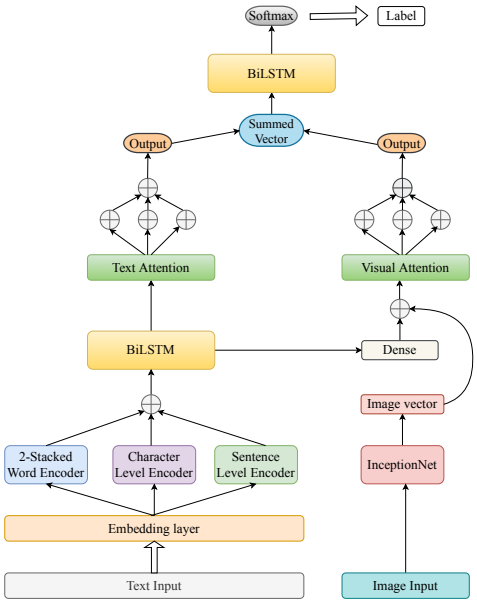

Figure 1: Multimodal Model

Textual Model

We define a BiLSTM-CNN model that takes the words, characters and sentence input into consideration. In order to incorporate the sentence-level representation s_o obtained in equation 3, we perform a `RepeatVector` operation to repeat the sentence vector over the maximum sequence length:

$$s_t = RepeatVector(s_o) \tag{4}$$

In this way, we repeat as a whole the context of the sentence over each word in the sequence. We then concatenate each representations in, 1,2 and 4 as follows: $m = [w_t; c_t; s_t]$ and feed this to a bidirectional LSTM layer followed by a softmax layer for the prediction given as followed:

$$\overrightarrow{h}_t = LSTM(m_i)$$
$$\overleftarrow{h}_t = LSTM(m_i)$$

In order to obtain the bidirectional representation of the input, we concatenated forward h_t and backward states h_t,

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{5}$$

Finally, we predict the probabilities over N classes as given below:

$$softmax(W_m h_t + b_m)$$

To this end, we share the methodology between our two approaches. And the multimodal approach is the extension of the textual model we already defined.

Multimodal Model

From previous works on multimodal named entity recognition, it is consistent that the auxiliary context from images better helps understanding microblogging text. (Esteves et al. 2018; Moon, Neves, and Carvalho 2018; Zhang et al. 2018). Therefore, we also incorporate visual information and propose improvement over existing attention-based approaches.

Segregated Contextual Attention. We propose a *segregated-contextual-attention-module* that considers text and image as separate modalities. Instead of computing a single context vector over naively merged textual and visual representations (e.g., $[h_t;v_i]$), we separately compute the textual and visual attention contexts as both modalities are considered as a whole. Then, we obtain the final context vector as a summed vector of both textual and visual attention context.

We observe that a word is only related to the small part or region of the image and incorporating the whole image with text may lead to suboptimal results consistent with the previous research in Zhang et al. (2018). The reason is more evident that the irrelevant regions would introduce noise, and at some point, the model will start to learn noise resulting in the worst results. To incorporate the visual context, we follow Zhang et al. (2018) and attend to those regions of the input image relevant to the text. To formulate visual attention, consider a given text h_t and image feature vector v_i. We first apply a `RepeatVector` operation over the original input image vector v_i and generate n copies, where n is equal to the maximum sentence length. We then, convert image feature vector v_i into an entirely new representation equal in the dimension of the text-vector by using a single layer perceptron as in (Zhang et al. 2018). The new image representation is given by,

$$v_i^{'} = \tanh(W_i v_i + b_i) \tag{6}$$

Similarly, we apply another `RepeatVector` operation to h_t and generate copies equal to the number of regions N in the image as we want to generate the text-relevant attention over the regions of the image. Thus, a new representation of text $h_t^{'}$ using a single-layer perceptron given by:

$$h_t^{'} = \tanh(W_{h_t} h_t + b_{h_t}) \tag{7}$$

Then, we concatenate each of the new representations as followed:

$$z_t = h_t^{'} \oplus v_i^{'}$$

Here, $\oplus$ denotes concatenation and z_t is the concatenation of text and image new representations. We then feed z_t to a neural network with (*tanh*) activation function as given below:

$$z_t = \tanh(W_{z_t} z_t + b_{z_t})$$
$$\alpha_t = \text{softmax}\left(W_{\alpha_t} z_t + b_{\alpha_t}\right)$$

Then, a visual context vector C_v as a result of a dot product of origin image representation v_i and attention weights α_t is given below:

$$C_v = \sum_i \alpha_{t,i} v_i$$

For computing text attention, we feed textual representation h_t to a single-layer neural network followed by a *softmax function* to generate attention weight probabilities over h_t as given below:

$$a_w = \tanh(W_{h_t} h_t + b_{h_t})$$
$$\alpha_t = \text{softmax}(W_{a_w} a_w + b_{a_w})$$

Textual context vector C_t is the weighted sum of dot products of attention weights α_t and text representation w_t given by,

$$C_t = \sum_i \alpha_{t,i} w_i$$

Approaches	WNUT 2016	WNUT 2017	CoNLL 2003 (Eng)	Multimodal
(Partalas et al. 2016)	46.16	-	-	-
(Limsopatham and Collier 2016)	52.41	-	-	-
(von Däniken and Cieliebak 2017)	-	40.78	-	-
(Zhang et al. 2018)	-	-	-	70.69
(Aguilar et al. 2019)	-	45.55	89.01	-
(Akbik, Bergmann, and Vollgraf 2019)	-	49.59	93.09	-
(Baevski et al. 2019)	-	-	93.5	-
(Wang et al. 2019)	-	50.03%	93.47	-
(Zhou et al. 2019)	53.43	-	-	-
Ours	**53.48**	**50.52**	**93.76**	**74.17**

Table 1: Results statistics over 4 datasets and comparison with the current state-of-the-art.

Combinations	WNUT 2016	WNUT 2017
2-Stacked Word BiLSTM	47.64 (Baseline)	47.27 (Baseline)
2-Stacked Word BiLSTM + Character	49.81	48.13
2-Stacked Word BiLSTM + Sentence	52.79	49.87
All features	53.48	50.52

Table 2: This table describe the importance of each type of feature in combination to the other.

Then, we compute the overall context vector C_m by taking the sum of textual and visual context vectors as depicted in Figure 1 as given below:

$$C_m = C_t + C_v$$

Finally, the overall context vector C_m was fed to a fully-connected bi-directional LSTM layer followed by a softmax as illustrated in the Figure 1 to predict the probabilities over the k classes :

$$BLSTM(C_m)$$

Incorporating only text-relevant context from images could reduce noise. And by leveraging maximum information gain from the two modalities helps in better representation of the input, hence, yields better results.

Experimentation

In this section, we will discuss the experimental setup and evaluation results of the system across.

Data

1. We use the WNUT 2016 (Strauss et al. 2016) data set comprised of the combination of train and development data originated from the work on the Twitter NER task (Ritter et al. 2011). The tokens were distributed over ten named entity types.
2. WNUT 2017 (Derczynski et al. 2017) is the second dataset we use in our work. The primary source of this data is Twitter. Additional sources include Reddit, YouTube, and StackExchange. The tokens were distributed over six entity types.
3. We also consider multimodal data from Zhang et al. (2018) to evaluate our multimodal approach. The dataset has four common entity types.
4. Finally, we use the CoNLL 2003 dataset (Tjong Kim Sang and De Meulder 2003), which is a collection of newswire articles from Reuters that also has four entity types.

Parameters. For the word-level BiLSTM, we set the number of units and dropout to 512 and 0.5 for each BiLSTM layer. The dropout helps the model to tackle overfitting (Srivastava et al. 2014). Following (Aguilar et al. 2017), we set the number of filters at each convolution layer to 64. We also set the sentence-level dense network with 256 hidden states. For the fully-connected BiLSTM layer, we set the number of units and dropout rate to 200 and 0.3, respectively. To reduce the potential overfitting, we also apply an L_2 weight regularization over convolutional and word-level BiLSTM layers with a rate of 10^{-3}. We optimize the parameters using RMSprop and apply early stopping technique with the patience of 3 epochs. We perform parameter search over the learning rate $\in \{10^{-3}, 10^{-2}, 10^{-1}\}$ and batch size $\in \{10, 20, 50\}$. We then select the best model with the highest F1-score on the validation set. After determining the parameters, we repeat the experiment 3 times with different random seeds, and train using both train and development set, reporting average performance on the test set as our final result. We used official WNUT script to compute the results [2].

Results and Discussion

WNUT 2016 results. One of the challenging aspects of the WNUT 2016 (Strauss et al. 2016) dataset is small training data. Moreover, there are 10 entity types instead of the usual 4 entities with overlapping entities like `person` and `musicartist`. Similarly, entities like `facility` and `company` are also at many occasions ambiguous. Often, a *sports-team* is named under the country or city it belongs

[2]https://noisy-text.github.io/2017/emerging-rare-entities.html

to, causing a considerable overlap with the geo-loc entity type. By contrast to earlier mentioned approaches, our system goes beyond the current state-of-the-art with 53.48% F1 score on WNUT 2016 without the need for external resources and additional data. Our benchmark with current systems is shown in Table 1.

WNUT 2017 results. The challenge that WNUT 2017 data (Derczynski et al. 2017) poses is the more frequent use of punctuation (Derczynski et al. 2014). Often, part of the name is also a common word (e.g., Andrew little) and some location names are also common person names (e.g., Smith), thus, causing difficulty (Derczynski et al. 2014). Entities like *corporation*, *group*, and *product* could easily be confused between each other. Current state-of-the-art, though do not rely on external resources, however, rely on the CRF based classifier. For instance, Aguilar et al. (2019) emphasized the significance of having a CRF prediction layer over *softmax* and reported an approximately 4% increase in F1 with CRF in comparison to a softmax. However, we found this claim not completely true since it may be dependent on the dataset. Consistent with our previous results, our model goes beyond the current state-of-the-art with an F1 score of 50.5% without relying on external resources and the CRF prediction layer. This improvement is not significant; nevertheless, it gave us a consistent result. We also observe the improvement over hard to predict entities such as creative-work and product compared to the current state-of-the-art.

Multimodal results. So far, we only consider the textual context. In this experiment, we evaluate our multimodal approach that considers the auxiliary context from the text's images. With *segregated attention network*, our model achieved an F1 score of 74.17% on this data which is a considerable improvement over the state-of-the-art model of (Zhang et al. 2018). In one of our experiments, we merged the textual and visual representation and observed results with and without attention. On both occasions, we received sub-optimal results mainly due to the noise pose by the images. Interestingly, considering textual and visual clues as entirely separate modalities and computing the final context vector as a summed vector over two separately computed attention contexts helped improve information gain and reduce the noise.

CoNLL-03 results. Lastly, to verify our model as a generalized solution for NER, we also experimented with a standard data set. Our evaluation results reveal that our model also goes beyond the state-of-the-art with 93.76% F1 score proving its suitability to the NER task.

Ablation and analysis. To reflect the usefulness of each feature combination we experimented with word embeddings only and observe 47.64% F1 score on WNUT 2016. We consider this result as a baseline and compare it with the results achieved by adding supplementary features to this baseline model. With word and character embeddings, the result improved by approximately 2% on WNUT 2017. Interestingly, we observed considerable improvement over baseline embedding with word and sentence-level information which is approximately 5% on WNUT 2016. However, not many previous works considered sentence-level information for the NER task. To the best of our knowledge, only

Nr.	Prediction
1	**Angels** manager Mike Scioscia said tuesday he's expect
2	I remember having parliament on the radio in my car and hearing **Leyonhjelm** [give this speach]
3	. **The Six Thatchers**
4	How **Miami Dolphins** defensive end spot stacks up ...
5	if **Leicester** wins the champions league and manu wins **Europa** league

Table 3: Sample predictions from WNUT 2016 and 2017 datasets.

von Däniken and Cieliebak (2017) employed sent2vec and reported negative results. In our case, sentence embedding from Universal-sentence-encoder in combination with word embedding improved our overall results. To verify this experimental finding, we performed another ablation experiment on a different data set. On WNUT 2017, we observe consistent improvement with the word and sentence feature combination as shown in Table 2. While Table 3 shows our model's sample predictions on WNUT 2016 and 2017 test sets. In the first example taken from the WNUT 2016 test dataset, our model can correctly classify ***'Angels'*** as sports-team without considering any external resources. It means that the network has based its prediction solely on the context. The second example from the WNUT 2017 test set has the word ***'Leyonhjelm'*** which is not an English word. Despite ***'Leyonhjelm'*** being an out-of-vocabulary word, our model can still correctly classify it as a person. In the third example, our model can correctly infer ***'The'*** belongs to the creative-work named entity which is ambiguous even for humans. Moreover, this is not the case for only one example; we observe many other examples where our model correctly classified ***the*** as the named entity part. According to Derczynski et al. (2017), WNUT 2017 data includes numerous entities that have either a common person name or a name of place counterpart. For example in Table 3, in ***'Miami Dolphins'*** the word ***'Miami'*** is not a name of the place.

Interestingly, our word-character model wrongly classify ***'Miami'*** as a location while our Word-sentence model correctly classifies it as a part of the group named entity. In another example, our model was correctly able to classify ***Leicester*** as a group. We believe that the sentence-level information helped in this case and the model predicted purely based on the sentence's whole context. In many other examples, we also observed that our model was correctly able to infer the label for the name of the place as the counterpart of the named entities such as group. Our model struggled with common names as person entity and at many occasion failed to predict the right label. For example, our model missed ***'Snowman'*** and ***'theguest'*** even though the context was there. Overall, product and corporation were the difficult classes to predict.

To this end, we conclude that word embeddings are the essential information for NER, and with word and sentence combination, we observe the most improvement. Our analysis also suggests that with the sentence-level information,

we see consistent improvement in proving its usefulness.

Conclusion

We presented two models for NER on microblog data. The first model considers solely textual information, while the second model also employs auxiliary information from images. Our evaluation suggests that our model can be successfully applied to NER in microblogs and does not require external sources, preprocessing, and employing a CRF classifier for sequence labelling. With our approach, we are not only able to go beyond the current state-of-the-art, but we are also able to address some of the weaknesses of the previous systems. Our evaluation also suggests that InferNER can be easily ported to other NER tasks without the need to alter supplementary resources.

References

Aguilar, G.; Maharjan, S.; López-Monroy, A.; and Solorio, T. 2017. A multi-task approach for named entity recognition in social media data. 148–153.

Aguilar, G.; López-Monroy, A. P.; González, F. A.; and Solorio, T. 2019. Modeling noisiness to recognize named entities using multitask neural networks on social media. *CoRR* abs/1906.04129.

Akbik, A.; Bergmann, T.; and Vollgraf, R. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 724–728. Minneapolis, Minnesota: Association for Computational Linguistics.

Baevski, A.; Edunov, S.; Liu, Y.; Zettlemoyer, L.; and Auli, M. 2019. Cloze-driven pretraining of self-attention networks. *CoRR* abs/1903.07785.

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strope, B.; and Kurzweil, R. 2018. Universal sentence encoder. *CoRR* abs/1803.11175.

Derczynski, L.; Maynard, D.; Rizzo, G.; Erp, M.; Gorrell, G.; Troncy, R.; Petrak, J.; and Bontcheva, K. 2014. Analysis of named entity recognition and linking for tweets. *Information Processing Management* 51:32–49.

Derczynski, L.; Nichols, E.; van Erp, M.; and Limsopatham, N. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 140–147. Copenhagen, Denmark: Association for Computational Linguistics.

Esteves, D.; Peres, R.; Lehmann, J.; and Napolitano, G. 2018. Named entity recognition in twitter using images and text. In Garrigós, I., and Wimmer, M., eds., *Current Trends in Web Engineering*, 191–199. Cham: Springer International Publishing.

Limsopatham, N., and Collier, N. 2016. Bidirectional LSTM for named entity recognition in twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, 145–152. Osaka, Japan: The COLING 2016 Organizing Committee.

Ma, X., and Hovy, E. H. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR* abs/1603.01354.

Moon, S.; Neves, L.; and Carvalho, V. 2018. Multimodal named entity recognition for short social media posts. *CoRR* abs/1802.07862.

Pagliardini, M.; Gupta, P.; and Jaggi, M. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *CoRR* abs/1703.02507.

Partalas, I.; Lopez, C.; Derbas, N.; and Kalitvianski, R. 2016. Learning to search for recognizing named entities in twitter. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, 171–177. Osaka, Japan: The COLING 2016 Organizing Committee.

Peres, R.; Esteves, D.; and Maheshwari, G. 2017. Bidirectional lstm with a context input window for named entity recognition in tweets. 1–4.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *CoRR* abs/1802.05365.

Ritter, A.; Clark, S.; Mausam; and Etzioni, O. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1524–1534. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Strauss, B.; Toma, B.; Ritter, A.; de Marneffe, M.-C.; and Xu, W. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, 138–144. Osaka, Japan: The COLING 2016 Organizing Committee.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2015. Rethinking the inception architecture for computer vision. *CoRR* abs/1512.00567.

Tjong Kim Sang, E. F., and De Meulder, F. 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*, 142–147.

von Däniken, P., and Cieliebak, M. 2017. Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 166–171. Copenhagen, Denmark: Association for Computational Linguistics.

Wang, Z.; Shang, J.; Liu, L.; Lu, L.; Liu, J.; and Han, J. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5154–5163. Hong Kong, China: Association for Computational Linguistics.

Zhang, Q.; Fu, J.; Liu, X.; and Huang, X. 2018. Adaptive co-attention network for named entity recognition in tweets. In *AAAI*.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2015. Learning deep features for discriminative localization. *CoRR* abs/1512.04150.

Zhou, J. T.; Zhang, H.; Jin, D.; Zhu, H.; Fang, M.; Goh, R. S. M.; and Kwok, K. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3461–3471. Florence, Italy: Association for Computational Linguistics.