Bidirectional LSTM for Named Entity Recognition in Twitter Messages

Nut Limsopatham and **Nigel Collier**
Language Technology Lab
Department of Theoretical and Applied Linguistics
University of Cambridge
Cambridge, UK
{nl347,nhc30}@cam.ac.uk

Abstract

In this paper, we present our approach for named entity recognition in Twitter messages that we used in our participation in the Named Entity Recognition in Twitter shared task at the COLING 2016 Workshop on Noisy User-generated text (WNUT). The main challenge that we aim to tackle in our participation is the short, noisy and colloquial nature of tweets, which makes named entity recognition in Twitter messages a challenging task. In particular, we investigate an approach for dealing with this problem by enabling bidirectional long short-term memory (LSTM) to automatically learn orthographic features without requiring feature engineering. In comparison with other systems participating in the shared task, our system achieved the most effective performance on both the 'segmentation and categorisation' and the 'segmentation only' sub-tasks.

1 Introduction

Named entity recognition (NER), which is one of the first and important stages in a natural language processing (NLP) pipeline, is to identify mentions of entities (e.g. persons, locations and organisations) within unstructured text. Traditionally, most of the effective NER approaches are based on machine learning techniques, such as conditional random field (CRF), support vector machine (SVM) and perceptrons (Lafferty et al., 2001; McCallum and Li, 2003; Settles, 2004; Luo et al., 2015; Ju et al., 2011; Ratinov and Roth, 2009; Segura-Bedmar et al., 2015). For instance, Ratinov and Roth (2009) effectively learned a perceptron model using features, including word classes induced using Brown clustering (Liang, 2005), and gazetteer extracted from Wikipedia.

Twitter NER is an NER task that aims to identify mentions of entities in Twitter messages (i.e. tweets) (Baldwin et al., 2015; Ritter et al., 2011). Twitter NER is particularly challenging because of the unique characteristics of tweets. For instance, tweets are typically short as the number of characters in a particular tweet is restricted to 140; hence, the contextual information is limited. In addition, the use of colloquial language makes it difficult for existing NER approaches a general domain, such as newswire to be reused (Baldwin et al., 2015). Consequently, state-of-the-art NER software (e.g. Standford NER) is less effective on Twitter NER tasks (Derczynski et al., 2015).

For our participation in the Named Entity Recognition in Twitter shared task at the COLING 2016 Workshop on Noisy User-generated text (WNUT) (Strauss et al., 2016), we aim to investigate a novel approach that allows neural network to explicitly learn and leverage orthographic features. We focus on orthographic features as they have shown to be effective and widely used in several NER systems. Importantly, orthographic features are used by majority of the systems (including the best system) participating in the Twitter NER shared task at the 2015 WNUT workshop (Baldwin et al., 2015).

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 145–152, Osaka, Japan, December 11 2016.

Social Media Sentence	Orthographic Sentence
14th MENA FOREX EXPO announced!!	nncc CCCC CCCCC CCCC ccccccccccpp
Nintendo 3DS released in north America	Cccccccc nCC cccccccc cc cccc Ccccccc
END!! Cowboys 28, Eagles 24	CCCpp Ccccccc nnp Cccccc nn
@NutLims c u 2mor	pCccCccc c c nccc
The Hobbit is on TV now!!	Ccc Cccccc cc cc CC cccpp

Table 1: Examples of social media sentences and their corresponding orthographic sentence.

2 Related Work

Conditional random field (CRF) is one of the most effective approaches (Lafferty et al., 2001; McCallum and Li, 2003; Settles, 2004) for NER, as it achieved state-of-the-art performances on several NER tasks, such as CoNLL03 (Tjong Kim Sang and De Meulder, 2003) or Twitter NER (Baldwin et al., 2015). In particular, CRF learns latent structures of an input sequence by using a undirected statistical graphical model. Nevertheless, the performance of CRF mainly depends on hand-crafted features designed specifically for a particular task or domain. Consequently, these hand-crafted features are difficult to develop and maintain. Examples of hand-crafted features are orthographic features (Bikel et al., 1999), which are based on patterns of characters contained in a given word. In this work, we investigate an approach for could automatically inducing and leveraging orthographic features for named entity recognition for Twitter messages.

Neural networks have recently shown to be effective for several NLP tasks, such as NER (Chiu and Nichols, 2015), POS tagging (Huang et al., 2015), sentiment analysis (Limsopatham and Collier, 2016b) and grounding (Limsopatham and Collier, 2016c; Limsopatham and Collier, 2015). For example, Collobert et al. (2011) designed a feed-forward neural network that learned to identify entities in a sentence by using contexts within a fixed number of surrounding words. Chiu and Nichols (2015) showed that modelling both character and word embeddings within a neural network for NER further improve the performance. Huang et al. (2015) introduced a more complex model based on bidirectional LSTM could also take into account hand-crafted features. In this work, we investigate an application of our novel approach (Limsopatham and Collier, 2016a) that enables bidirectional LSTM to automatically induce orthographic features rather than feeding hand-crafted features into the model, when performing Twitter NER.

3 Bidirectional LSTM for Twitter NER

In this section, we describe our end-to-end neural network approach for Twitter NER. In particular, our approach consists of three main components: (1) orthographic sentence generator, (2) word representations as input vectors, (3) bidirectional LSTM.

3.1 Orthographic Sentence Generator

Our orthographic sentence generator creates an *orthographic sentence*, which contains orthographic pattern of words in each input sentence. In particular, for a given social media sentence (e.g. '14th MENA FOREX EXPO announced!!'), we generate an orthographic sentence (e.g. 'nncc CCCC CCCCC CCCC ccccccccccpp') by using a set of rules, where each of the upper-case characters, lower-case characters, numbers and punctuations, are replaced with *C*, *c*, *n* and *p*, respectively. Examples of orthographic sentences generated from social media sentences are shown in Table 1. This orthographic sentence allows bidirectional LSTM to explicitly induce and leverage orthographic features automatically.

3.2 Input Vectors for Bidirectional LSTM

Our approach uses word representations extracted from both character and word levels. To do so, we create vectors of character-based word representation (Section 3.2.1) and word representation (Section 3.2.2) for both social media sentence and its orthographic sentence, as follows:

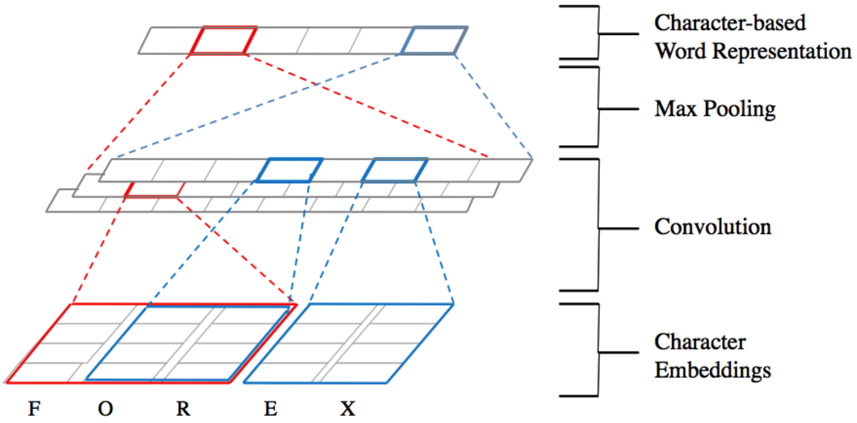

Figure 1: Our CNN architecture for inducing word representation from a character level.

3.2.1 Character-based Word Representation

We use CNN to induce word representation from character embeddings of a given word, as shown in Figure 1. Specifically, for a given word of length l characters, we create a word matrix $\mathbf{M} \in \mathbb{R}^{d \times l}$ as:

$$\mathbf{M} = \begin{bmatrix} | & | & | & | & \\ \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_l \\ | & | & | & | & \end{bmatrix} \tag{1}$$

where each column of $\mathbf{M}$ is the d-dimensional vector (i.e. character embedding) $\mathbf{x}_i \in \mathbb{R}^d$ of each character in the given word, which are initialised randomly.

To learn patterns of characters in a given word, a convolution operation with a filter $\mathbf{w} \in \mathbb{R}^{d \times h}$ is applied to a window of h characters. In particular, a filter $\mathbf{w}$ is convolved over the sequence of characters in the word matrix $\mathbf{M}$, which results in a feature matrix $\mathbf{C}$. Specifically, each feature c_i in feature matrix $\mathbf{C}$ is extracted from a window of words $\mathbf{x}_{i:i+h-1}$, as follows:

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \tag{2}$$

where f is an activation function, e.g. sigmoid and tanh, and $b \in \mathbb{R}$ is a bias. In this work, we use 200 different filters with window size $h = 3$.

Then, we follow Collobert et al. (2011) and apply max pooling to capture the most important feature from each filter. Indeed, max pooling takes the maximum value of each row in the matrix $\mathbf{C}$:

$$\mathbf{c}_{max} = \begin{bmatrix} max(\mathbf{C}_{1,:}) \\ \vdots \\ max(\mathbf{C}_{d,:}) \end{bmatrix} \tag{3}$$

We use $\mathbf{c}_{max}$ vector as a character-based word representation in bidirectional LSTM (Section 3.3), as it captures important features of a given word induced from a character level.

3.2.2 Word Representation

Existing studies, e.g. (Mikolov et al., 2013; Pennington et al., 2014), have shown that word embeddings induced from a large corpus could effectively capture semantic and syntactic information of words. Hence, we also use pre-trained word embeddings as word representation. However, any randomly generated word embeddings can also be used.

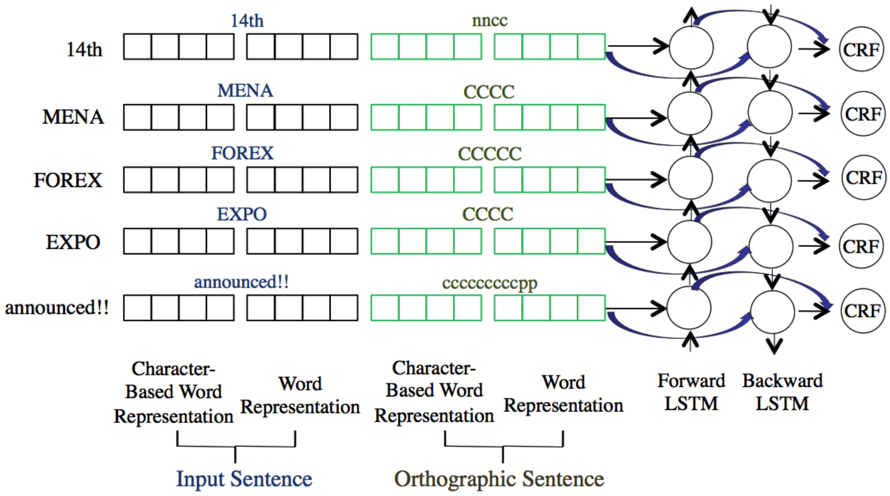

Figure 2: Our bidirectional LSTM for Twitter NER.

3.3 Bidirectional LSTM

In this work, we use bidirectional LSTM for modelling social media sentences, as existing work (e.g. (Huang et al., 2015; Dyer et al., 2015; Dyer et al., 2015; Bengio et al., 1994)) has shown that bidirectional LSTM could effectively deal with the variable lengths of sentences. In addition, it could capture past (from the previous words) and future (from the next words) information effectively (Huang et al., 2015; Dyer et al., 2015).

Our bidirectional LSTM for Twitter NER is shown in Figure 2. For a given a social media sentence and its orthographic sentence, we firstly extract both character-based word representation and word vector representation corresponding to each word in the social media sentence and the orthographic sentence, by using the approaches previously described in Sections 3.2.1 and 3.2.2.[1] Word representations associated to the same words are then concatenated and sequentially fed into bidirectional LSTM to learn contextual information of words in the sentence. At the output layer, we optimise the CRF log-likelihood, which is the likelihood of labelling the whole sentence correctly by modelling the interactions between two successive labels using the Viterbi algorithm, as suggested by Huang et al. (2015).

4 Experimental Setup

4.1 Datasets

The Twitter NER shared task datasets consist of *training set* (i.e. '2015 train'+'2015 dev'), *development set* (i.e. '2015 test'), *additional set* (i.e. additional 'dev 2015') and *test set*, respectively. The numbers of tweets and tokens of each set are shown in Table 2. The shared task focuses on finding 10 types of target entities, including company, facility, geo-location, movie, music-artist, other, person, product, sport team and TV show. In particular, the shared task can be divided to to sub-tasks: 'segmentation only' and 'segmentation and categorisation'. The former focuses only on finding the boundaries of entities; meanwhile, the latter requires both the boundaries of entities and the correct categories of entity types.

4.2 Training Regime

To learn bidirectional LSTM, we use only the four datasets (see Table 2) provided by the workshop organisers. In particular, we use the combination of *training set* and *development set* as training data, and use *additional set* as validation data, when generating the test models. Note that we train two different models for the two sub-tasks, i.e. 'segmentation and categorisation' (10-type) and 'segmentation only' (no-type).

[1] Note that we use separated set of word and character embeddings for the input sentence and the orthographic sentence.

	Training Set	Development Set	Additional Set	Test Set
# tweets	2,349	1,000	419	3,850
# tokens	46,469	16,261	6,789	61,908
# entity tokens	2,462	1,128	439	5,955
# Company entity tokens	207	49	64	886
# Facility entity tokens	209	77	13	619
# Geo-loc entity tokens	325	158	62	1,101
# Movie entity tokens	80	30	5	82
# MusicArtist entity tokens	116	76	22	331
# Other entity tokens	545	229	91	1,140
# Person entity tokens	664	266	113	782
# Product entity tokens	177	158	15	746
# SportsTeam entity tokens	74	83	46	195
# TvShow entity tokens	65	2	8	73

Table 2: Statistics of the WNUT 2016 NER shared task datasets.

To tune hyper-parameters of the model, we optimise the performance on the *development set* when training on the *training set*.

4.3 Embeddings

4.3.1 Word Embeddings

As discussed in Section 3.2.2, our approach uses word embeddings as inputs when learning an NER model. For input sentences, we use pre-trained word embeddings of Godin et al. (2015)[2], which consists 400-dimensional vectors of 3.04 million unique words induced from 400 million Twitter messages using the Skip-gram model from the word2vec tool (Mikolov et al., 2013). For the words that do not exist in the pre-trained embeddings, we use a vector of random values sampled from $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$ where dim is the dimension of embeddings as suggested by He et al. (2015).

For orthographic sentences, we represent each word using a 200-dimensional randomly generated vector, where each dimension is also uniformly sampled from $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$.

4.3.2 Character Embeddings

We use 30-dimensional character embeddings for representing each character when inducing the character-based word representation (Equation (1) in Section 3.2.1) from both social media and orthographic sentences. The 30-dimensional character embeddings are initialised using uniform samples from $[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}]$. Note that we have a separated embedding for each set of characters in the social media and orthographic sentences.

4.4 Parameter Optimisation

We implement our bidirectional LSTM using the Theano library (Bergstra et al., 2010). Parameter optimisation is done by mini-batch stochastic gradient descent (SGD) where back-propagation is performed using Adadelta update rule (Zeiler, 2012). The mini-batch size is 50. In addition, we allow the learner to fine-tune both word and character embeddings when performing gradient updates during training. Moreover, we follow Pascanu et al. (2013) and use a gradient clipping of 5.0, in order to reduce gradient exploding.

To avoid overfitting, we apply L_2 regularisation on the weight vectors and dropout (Srivastava et al., 2014) on hidden units in all layers in our models. Dropout rate is set to 0.5. We also use early stopping (Giles, 2001) based on the performance achieved on the development sets.

[2] Downloaded from `http://www.fredericgodin.com/software`.

Approach	Segmentation and Categorisation (10-Type)			Segmentation Only (No-Type)		
	F1	Precision	Recall	F1	Precision	Recall
Our approach	**52.41**	**60.77**	**46.07**	**65.89**	**73.49**	59.72
Talos	46.16	58.51	38.12	60.24	70.53	52.58
akora	44.77	51.70	39.48	59.05	64.75	54.28
NTNU	40.06	53.19	32.13	63.22	64.18	**62.28**
ASU	39.02	40.58	37.58	55.17	57.55	52.98
DeepNNNER	37.24	54.97	28.16	47.82	70.66	36.14
DeepER	36.95	45.40	31.15	51.38	63.17	43.31
hjpwhu	36.22	48.90	28.76	46.66	63.00	37.06
UQAM-NTL	29.82	40.73	23.52	44.30	53.21	37.95
LIOX	19.26	40.15	12.69	40.73	58.18	31.33

Table 3: Performances in terms of F1, precision and recall of our approach and the participating systems on the 'segmention and categorisation' (10-type) and the 'segmentation only' (no-type) sub-tasks.

Type	F1	Precision	Recall
company	57.22	69.84	48.47
facility	42.42	51.70	35.97
geo-loc	72.61	75.21	70.18
movie	10.91	14.29	8.82
musicartist	9.48	26.83	5.76
other	31.66	49.45	23.29
person	58.99	52.06	68.05
product	20.12	36.96	13.82
sportsteam	52.41	53.15	51.70
tvshow	5.88	100.00	3.03
Overall	52.41	60.77	46.07

Table 4: Performances of our approach broken down by entity types.

5 Experimental Results

Next, we discuss the performance of our proposed approach. Table 3 compares the performances of our approach with the other systems participating in the Twitter NER shared task at the 2016 WNUT workshop, in terms of F1, precision and recall measures, on both the 'segmentation and categorisation' and 'segmentation only' sub-tasks.

From Table 3, we observed that our approach achieved the best F1 score for both sub-tasks. In particular, our approach attains F1 scores of 52.41 and 65.89 for the 'segmentation and categorisation' and 'segmentation only' sub-tasks, respectively. Importantly, for the 'segmentation and categorisation' sub-task, our approach significantly outperformed the second best system (namely, Talos) by 6.2 F1 score. Table 4 showed the performance of our approach broken down into entity types. Our approach performed effectively on entities related to *geo-location*, *person* and *company*. Meanwhile, it was less effective on the entity types *tvshow*, *musicartist* and *movie*.

6 Conclusions

In this paper, we describe our novel approach used in the Twitter NER shared task at the WNUT 2016 workshop. Our approach deals with the noisy and colloquial of tweets by leveraging word representations of input sentence and orthographic sentence in bidirectional LTSM. In particular, our approach automatically induce and leverage orthographic features when performing NER. Importantly, we show that without requiring hand-crafted features, our approach is highly effective for the Twitter NER tasks, as it achieves the best performance among all of the participating systems. For future work, we aim to investigate approaches for enabling neural networks to automatically induce other hand-crafted features, such as gazetteers.

Acknowledgements

The authors wish to thank funding support from the EPSRC (grant number EP/M005089/1).

References

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.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*.

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what&lsquo;s in a name. *Mach. Learn.*, 34(1-3):211–231, February.

Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

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.

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*.

Rich Caruana Steve Lawrence Lee Giles. 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, volume 13, page 402. MIT Press.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab@ acl w-nut ner shared task: Named entity recognition for twitter microposts using distributed word representations. *ACL-IJCNLP 2015*, page 146.

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.

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

Zhenfei Ju, Jian Wang, and Fei Zhu. 2011. Named entity recognition from biomedical text using svm. In *Bioinformatics and Biomedical Engineering,(iCBBE) 2011 5th International Conference on*, pages 1–4. IEEE.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Nut Limsopatham and Nigel Collier. 2015. Adapting phrase-based machine translation to normalise medical terms in social media messages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1675–1680, Lisbon, Portugal, September. Association for Computational Linguistics.

Nut Limsopatham and Nigel Collier. 2016a. Learning orthographic features in bi-directional lstm for biomedical named entity recognition. In *Proceedings of the 2016 biennial Workshops on Building and Evaluating Resources for Biomedical Text Mining*. Association for Computational Linguistics.

Nut Limsopatham and Nigel Collier. 2016b. Modelling the combination of generic and target domain embeddings in a convolutional neural network for sentence classification. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics.

Nut Limsopatham and Nigel Collier. 2016c. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1014–1023, Berlin, Germany, August. Association for Computational Linguistics.

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, Lisbon, Portugal, September. Association for Computational Linguistics.

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 at HLT-NAACL 2003-Volume 4*, pages 188–191. 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 *NIPS*, pages 3111–3119.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

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*, pages 1524–1534. Association for Computational Linguistics.

Isabel Segura-Bedmar, Vıctor Suárez-Paniagua, and Paloma Martınez. 2015. Exploring word embedding for drug name recognition. In *SIXTH INTERNATIONAL WORKSHOP ON HEALTH TEXT MINING AND INFORMATION ANALYSIS (LOUHI)*, page 64.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics.

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

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

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-Volume 4*, pages 142–147. Association for Computational Linguistics.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.