DeepNNNER: Applying BLSTM-CNNs and Extended Lexicons to Named Entity Recognition in Tweets

Fabrice Dugas
École Polytechnique de Montréal
faberice.dugas@gmail.com

Eric Nichols
Honda Research Institute Japan Co.,Ltd.
e.nichols@jp.honda-ri.com

Abstract

In this paper, we describe the DeepNNNER entry to The 2nd Workshop on Noisy User-generated Text (WNUT) Shared Task #2: Named Entity Recognition in Twitter. Our shared task submission adopts the bidirectional LSTM-CNN model of Chiu and Nichols (2016), as it has been shown to perform well on both newswire and Web texts. It uses word embeddings trained on large-scale Web text collections together with text normalization to cope with the diversity in Web texts, and lexicons for target named entity classes constructed from publicly-available sources. Extended evaluation comparing the effectiveness of various word embeddings, text normalization, and lexicon settings shows that our system achieves a maximum F1-score of 47.24, performance surpassing that of the shared task's second-ranked system.

1 Introduction

Named entity recognition (NER) is an important part of natural language processing. It is a challenging task that requires robust recognition to detect common entities over a large variety of expressions and vocabularies. These problems are intensified when targeting Web texts because of challenges such as differences in spelling and punctuation conventions, neologisms, and Web markup (Baldwin et al., 2015).

Traditional approaches to NER on newswire texts has been dominated by machine learning methods that rely heavily on manual feature engineering and external knowledge sources (Ratinov and Roth, 2009; Lin and Wu, 2009; Passos et al., 2014). Recently, neural network models – especially those that use recursive models – have shown that state of the art performance can be achieved with little feature engineering (Collobert et al., 2011; Santos et al., 2015; Chiu and Nichols, 2016). However, despite their popularity for NER on newswire texts, neural networks have not been widely adopted for NER on Web texts, with the exception of the feed-forward neural network (FFNN) model of Godin et al. (2015).

In this paper, we present the DeepNNNER entry to the WNUT 2016 Shared Task #2: Named Entity Recognition in Twitter. Our shared task submission is based on the model of Chiu and Nichols (2016), a hybrid model of bidirectional long short-term memory (BLSTM) networks and convolutional neural networks (CNN) that automatically learns both character- and word-level features, and which holds the current state-of-the-art on both newswire texts (CoNLL 2003) and diverse corpora including Web texts (OntoNotes 5.0). In contrast to CRFs, FFNNs, and other windowed models, the BLSTM gives our model effectively infinite context on both sides of a word during sequential labeling. The character-level CNN allows our model to learn relevant features from the orthography of words, which is important in task where unseen words are commonplace. Finally, it also encodes partial lexicon matches in neural networks, allowing it to make effective use of lexical knowledge.

Our primary contribution is adapting the model of Chiu and Nichols (2016) to Twitter data by developing a text normalization method to effectively apply word embeddings to large vocabulary Web texts and automatically constructing lexicons for the shared task's target NE classes from publicly-available sources. The rest of our paper is organized as follows. In Section 2, we describe the adaptations made to Chiu and Nichols (2016)'s model. In Section 3, we describe the evaluation methodology. In Section 4, we discuss the results and present an error analysis. In Section 5, we summarize related research. Finally, in Section 6, we give concluding remarks.

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 178–187, Osaka, Japan, December 11 2016.

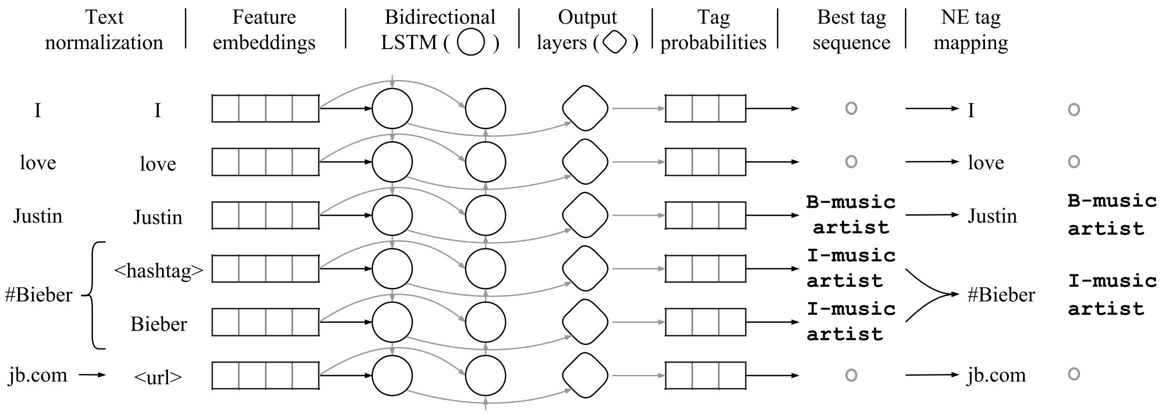


Figure 1: Our proposed system architecture for NER. Feature embeddings are constructed following Section 2.1. The output from both the forward and the backward LSTM are fed through a linear and a log-softmax layer before being added together (shown as ”Output layers”) to produce the tag scores.

2 Model

In this section, we describe the architecture of our shared task submission. An overview is given Figure 1. Our system is based on the BLSTM-CNN model of Chiu and Nichols (2016), and, unless otherwise noted, follows their training and tagging methodology, which the reader is referred to for more details.

2.1 Features

Feature embeddings for words are constructed by concatenating together the features listed here.

2.1.1 Word Embeddings

Word embeddings are critical for high-performance neural networks in NLP tasks (Turian et al., 2010). In this paper, we compare six publicly available pre-trained word embeddings. The embeddings are described in detail in Table 3. The neural embeddings of Collobert et al. (2011) were chosen because Chiu and Nichols (2016) reported them to be the highest performing on both CoNLL-2003 and OntoNotes 5.0 datasets. To evaluate embeddings trained on data closer to the WNUT dataset, we also selected the GloVe embeddings of Pennington et al. (2014), trained on both Web text and tweets, and word2vec embeddings trained on Google News data (Mikolov et al., 2013) and on tweets (Godin et al., 2015).

Preliminary evaluation on the Dev1 data showed that GloVe 27B outperformed Collobert's embeddings (see Table 5) and word2vec 3B, so they were used in our submission. Following Collobert et al. (2011), we use lookup tables to extract embeddings and every word is lower cased before lookup.

2.1.2 CNN-extracted Character Features

Following Chiu and Nichols (2016), we use a CNN to extract features from 25 dim. character embeddings randomly-initialized from a uniform distribution between -0.5 and 0.5. To accommodate text normalization, we added embeddings for the normalization symbols described in Section 2.2, namely `<url>`, `<user>`, `<smile>`, `<lolface>`, `<sadface>`, `<neutralface>`, `<heart>`, `<number>` and `<hashtag>`. All experiments were conducted with the same character embeddings.

2.1.3 Lexicon Features

Prior knowledge in the form of lexicons (also known as “gazetteers”) has been shown to be essential to NER (Ratinov and Roth, 2009; Passos et al., 2014). This section describes how the lexicons employed by our system were constructed. We designed the lexicon categories to be as close as possible to the shared task NE classes by extracting corresponding descendants from the DBpedia ontology (Auer et al., 2007). The lexicon used by our system contains 2.2 million entries over 9 different categories, as shown in Table 1. While most of the lexicons were extracted using only one descendant from the ontology, `Misc`, `Music`, and `Product` were constructed using multiple classes.

Text	Steve	King	on	ObamaCare	and	Constitution	Day
Company	–	–	–	–	–	–	–
Location	–	S	–	–	–	–	–
Misc	–	–	–	–	–	B	E
Movie	–	–	–	–	–	–	–
Music	–	–	–	–	–	–	–
Person	B	E	–	–	–	–	–
Product	–	–	–	–	–	–	–
SportsTeam	–	–	–	–	–	–	–
TVShow	–	–	–	–	–	–	–
WNUT Gold	B-person	E-person	O	O	O	B-other	E-other

Figure 2: Example of lexicon partial matching. The BIE tags indicate whether the token matched the beginning, inside or end of a lexical entry, and the S tag indicate an exact match with a single entry.

Category	**Entries**	**DBpedia Classes**
Company	57,856	Company
Location	710,704	Place
Misc	132,055	Work + Event - Movie - TVShow
Movie	87,234	Movie
Music	74,816	Band + MusicalArtist
Person	1,074,384	Person
Product	37,820	Device + WNUT:Product
SportsTeam	28,155	SportsTeam
TVShow	29,272	TelevisionShow
Total	2,232,295	

Table 1: Number of entries for each lexical category and their corresponding DBpedia classes.

Hyper-parameter	**Final**	**Range**
Convolution width	**5**	$[3, 9]$
CNN output size	**51**	$[15, 85]$
LSTM state size	**200**	$[100, 525]$
LSTM layers	**1**	$[1, 5]$
Learning rate	**0.0138**	$[10^{-3}, 10^{-1.8}]$
Epochs	**50**	-
Dropout	**0.56**	$[0.25, 0.75]$
Mini-batch size	**9**	-

Table 2: Hyper-parameter search space and final values used for all experiments.

First, in order to match entries such as festivals, holidays, songs, and more from the other class, we constructed the Misc lexicon from Event and Work types in the DBpedia ontology excluding Movie and TelevisionShow to avoid overlap with other classes. Second, in order to deal with inconsistencies between person and musicartist classes as discussed in Section 3, the Music lexicon is a combination of the subtypes Band and MusicalArtist[1]. Finally, in order to maximize coverage, the Product lexicon is a combination of the subtype Device from the DBpedia ontology and the lexicon *product* distributed with WNUT dataset. Every other category is as described in Table 1.

To generate lexicon features, we apply the partial matching algorithm of Chiu and Nichols (2016) to the input text, as shown in Figure 2. Each lexicon and match type (BIOES) is associated with a randomly-initialized 5 dim. embedding. The embeddings for all lexicons are concatenated together to produce the lexicon feature for each word in the input. To facilitate matching, all entries were stripped of parentheses and tokenized with the Penn Treebank tokenization script.

2.1.4 Capitalization Feature

Following Chiu and Nichols (2016), we used different symbols for word-level capitalization feature each assigned a randomly initialized embedding: allCaps, upperInitial, lowercase, mixedCaps and noinfo. Similar symbols were used for character-level (upper case, lower case, punctuation, other).

2.2 Text Normalization

In order to maximize word embedding lookup coverage, we modify the publicly available GloVe preprocessing script[2] to normalize irregular spelling and replace special symbols with special embeddings: <url>, <user>, <smile>, <lolface>, <sadface>, <neutralface>, <heart>, <number> and <hashtag>. Repeated punctuation is also removed.

When processing hashtags, the hashtag body is split on capital letters, distributing the NE tag across the resulting tokens. This helps increase word embedding coverage. Refer to Figure 1 for an example.

[1] Because MusicalArtist is a subtype of Person, both lexicons overlap, but experiments showed that the system still performed better when MusicalArtist was in both lexicons.

[2] http://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

Word Embeddings	Data (Size)	Dims.	Vocab.	Types		Tokens	
				-	+norm.	-	+norm
Collobert	RCV1 + Wikipedia (850M words)	50	130K	51.43%	75.31%	84.63%	82.18%
GloVe 6B	Gigaword5 + Wikipedia (6B words)	50	400k	54.22%	82.71%	86.02%	84.17%
GloVe 27B	Twitter microposts (27B words)	50	1.2M	57.47%	90.47%	83.67%	**97.66%**
GloVe 42B	Common Crawl (42B words)	300	1.9M	62.14%	90.86%	**89.09%**	86.21%
word2vec 3B	Google News (3B words)	300	3M	55.65%	86.00%	74.95%	72.62%
word2vec 400MT	Twitter microposts (400M *tweets*)	400	3M	**63.36%**	**91.08%**	87.23%	85.79%

Table 3: A comparison of word embedding type and token coverage with and without text normalization.

Additionally, we attempted to correct the most obvious spelling irregularities where letters in a word are repeated more than twice, consulting a dictionary to decide wether to keep one or two occurrences of that repeated letter. When consulting the dictionary, we prioritized shorter matches when the repeated letter appeared at the end of the word and longer matches otherwise.

For evaluation of the final system, we mapped the NE tags onto the original test data tokens, as shown in Figure 1. Because of the tokenization, some of the original entries could end up with more than one tag. In this case, we prioritize entity over non-entity tags, and keep the most frequent tag. Prioritizing entity over non-entity tags was meant to improve recall, albeit at the expense of precision.

Initial experiments on Dev1 comparing word2vec 3B, Collobert, and GloVe 27B embeddings showed that text normalization improved performance for word2vec 3B and GloVe 27B but not Collobert[3] (Table 5); that word type coverage increased drastically for all embeddings; and that while word token coverage greatly increased for GloVe 27B, it slightly decreased for other embeddings (see Table 4). We thus selected GloVe 27B embeddings for our submission due to their superior performance and coverage.

2.3 Training and Inference

We follow the training and inference methodology of Chiu and Nichols (2016), training our neural network to maximize the sentence-level log-likelihood from Collobert et al. (2011). Training is done by mini-batch SGD with a fixed learning rate, and we apply dropout (Pham et al., 2014) to the output nodes. All feature representations are “unfrozen” and allowed to be updated by the training algorithm.

We used the IOB tag scheme to annotate named entities. We also explored the BIOES tag scheme[4], as it was reported to outperform IOB (Ratinov and Roth, 2009), however, IOB outperformed BIOES in preliminary experiments. We suspect that data sparsity prevented the model from learning meaningful representations for the extra tags. Our shared task submission’s model trained in approximately 90 minutes and tags the test set in approximately 20 seconds, with memory usage peaking at 350MB[5].

2.4 Hyper-parameter Optimization

To maximize performance, we perform hyper-parameter optimization using Optunity’s implementation of particle swarm (Claesen et al., 2014), as there is some evidence that it is more efficient than random search (Clerc and Kennedy, 2002). The hyper-parameters of our model and final selected values are given in Table 2. We evaluated 800 hyper-parameter settings in total. The search used 5-fold validation to maximize the influence of the entire dataset, as it was small, and we kept the best performing setting.

3 Evaluation

The WNUT 2016 dataset consists of user-generated tweets tagged with 10 types of named entities: company, facility, geo-loc, movie, musicartist, other, person, product, sportsteam, and tvshow. Table 4 shows the train, dev and test set data splits. Compared to the well-researched CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) or the OntoNotes 5.0 dataset (Pradhan et al., 2013), the WNUT dataset contains a lot of spelling irregularities and special symbols. For example, *Christmas* written as *xmas*, *Guys* written as *Gaiiissss*, emoticons such as *“:-)”, “:(”, “<3”*

[3] Neither difference was statistically significant.

[4] BIOES stands for Begin, Inside, Outside, End, Single, indicating the position of the token in the entity.

[5] Our models were trained on a 2.7 GHz 12-Core Intel Xeon E5 CPU, with a maximum of 4 cores being used at once.

	Train	Dev1	Dev2	Test
Sentences	1,795	599	420	3,850
Tokens	34,899	11,570	6,790	61,908
Entities	1,140	356	272	3,473

Table 4: The WNUT 2016 dataset.

Settings	**Dev1**		
	Prec.	**Recall**	**F1**
Collobert	**56.92**	47.33	51.63 (±2.11)
Collobert + norm	55.80	45.84	50.24 (±1.52)
GloVe 27B	54.89	49.07	51.78 (±1.91)
GloVe 27B + norm	54.41	**49.86**	**51.96** (±1.25)
word2vec 3B	54.21	46.74	50.08 (±2.98)
word2vec 3B + norm	54.94	47.44	50.81 (±2.42)

Table 5: Dev1 preliminary evaluation.

Model	**Test**			
	Prec.	**Recall**	**F1**	**Rank***
CambridgeLTL	**60.77**	**46.07**	**52.41**	**1**
Talos	58.51	38.12	46.16	2
akora	51.70	39.48	44.77	3
NTNU	53.19	32.13	40.06	4
ASU	40.58	37.58	39.02	5
Submitted model	**54.97**	28.16	37.24	6
*Fixed model	52.04	39.63	44.97	3
**Best model	54.02	**42.06**	**47.24**	**2**

Table 6: F1-scores for our submitted, fixed, and best 10 tag models. *Rank** is the retroactive rank.

Settings	**Test**		
	Prec.	**Recall**	**F1**
Collobert	50.52	38.00	43.37 (±0.47)
+ norm	51.44	38.13	43.73 (±0.66)
+ lex	51.50	39.73	44.82 (±0.79)
+ norm + lex	51.72	40.46	45.39 (±1.15)
GloVe 6B	50.26	38.90	43.84 (±1.08)
+ norm	52.21	38.10	43.96 (±1.19)
+ lex	51.14	38.95	44.19 (±0.63)
+ norm + lex	51.73	38.73	44.24 (±0.44)
GloVe 27B	48.44	38.44	42.86 (±1.06)
+ norm	49.35	39.66	43.92 (±0.99)
+ lex	51.13	39.12	44.31 (±1.14)
*+ norm + lex	52.04	39.63	44.97 (±0.65)
GloVe 42B	51.81	40.87	45.59 (±0.71)
+ norm	52.91	41.60	46.53 (±0.84)
+ lex	52.27	41.50	46.22 (±0.93)
**+ norm + lex	**54.02**	**42.06**	**47.24** (±0.70)
word2vec 3B	51.71	38.56	44.11 (±0.47)
+ norm	53.92	37.82	44.39 (±1.28)
+ lex	51.37	39.01	44.31 (±0.68)
+ norm + lex	52.64	39.53	45.10 (±0.91)
word2vec 400MT	50.62	40.92	45.22 (±0.76)
+ norm	51.95	40.41	45.45 (±0.76)
+ lex	53.23	41.45	46.59 (±0.92)
+ norm + lex	53.87	41.78	47.03 (±0.97)

Table 7: F1-scores with different word embeddings evaluated on test set with final settings.

and so on are commonplace. Such examples illustrate the diversity of the dataset's vocabulary, motivating us to perform text normalization as described in Section 2.2.

Some inconsistencies were found between Dev2 and the other data. The most obvious one is where singers previously tagged as `person` in Train were tagged as `musicartist` in Dev2. This is easily verifiable by comparing tags for the entity *Justin Bieber* in those datasets. These tag inconsistencies make it difficult to learn a robust model for those classes, so we manually retagged all person entities, keeping the most precise tag (i.e. tagging all singers as `musicartist`). We did so by searching for every person entity with Google and used the surrounding context to determine the most precise tag, replacing a total of 82 `person` entities out of 664. Other local inconsistencies were not corrected as not enough evidence was found. In Section 4.3.2 we explore inconsistencies in common tagging errors.

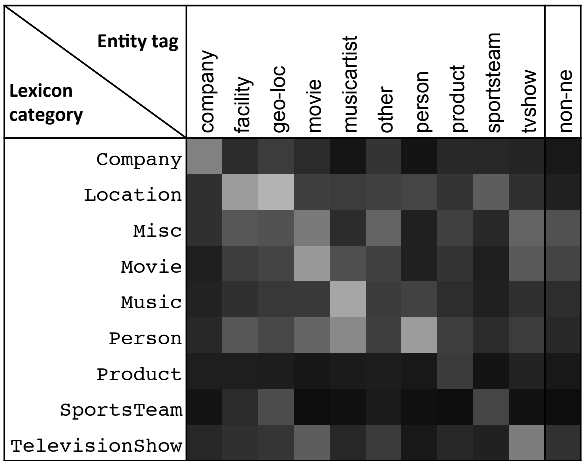
For each experiment, we report the average for precision and recall, and the average and standard deviation for f1-score for 10 successful trials. Minor inconsistencies in reported f1-scores and precision and recall result from those scores being averaged independently. Statistical significance is calculated using the Wilcoxon rank sum test, due to its robustness against small sample sizes with unknown distributions.

4 Results and Discussion

In this section, we (1) compare the performance of different word embeddings, (2) analyze the influence of our lexicon over the performance of our final model, and (3) perform error analysis of various aspects of both our system and the dataset. Table 7 shows the final results for different settings. Following Cherry et al. (2015), we compare our system settings to other shared task entries (Strauss et al., 2016) and present their retroactive ranks. While our submitted system uses GloVe embeddings trained on Twitter (GloVe 27B), we found that GloVe embeddings trained on Common Crawl (GloVe 42B) with text normalization and lexicons was our best performing setting, achieving a retroactive rank of second place.

4.1 Word Embeddings

Table 7 shows that GloVe embeddings trained on Common Crawl (GloVe 42B) outperformed all other embeddings by over 2 f1 points. Comparing Tables 4 and 7, we see that word type coverage is correlated



Predicted Tag / Gold Tag	company	facility	geo-loc	movie	music…	other	person	product	Sports…	tvshow	O
company	328	17	**48**	2	1	**75**	25	15	3	0	**452**
facility	19	234	**101**	0	5	22	43	22	0	1	**194**
geo-loc	9	**41**	770	0	0	12	**45**	0	8	1	**355**
movie	1	0	0	5	2	**10**	1	0	1	4	**72**
musicartist	0	2	0	0	50	31	**112**	5	0	0	**177**
other	39	23	**64**	3	6	311	31	7	11	6	**772**
person	4	0	5	1	**14**	6	606	0	0	0	**169**
product	13	0	6	5	24	**57**	28	88	5	0	**650**
sportsteam	4	3	**42**	0	2	17	7	0	69	0	**96**
tvshow	0	4	3	2	0	**6**	2	1	0	7	**64**
O	50	34	66	1	19	95	61	40	27	2	32207

Figure 3: Left: fraction of named entities of each class matched by entries in each lexicon category. White = higher fraction. Right: entity-level confusion matrix with outliers highlighted.

with performance for GloVe and word2vec embeddings[6]. Note that word type coverage appears to be more important than token coverage. It could be the case that NEs are more likely to contain low-frequency words, necessitating a large token vocabulary. GloVe 42B's increased performance over GloVe 27B could be explained thus, though it is also possible that its larger, more diverse dataset is responsible.

It is interesting to point out that Collobert was able to outperform embeddings trained on much larger datasets with much larger vocabularies. While all embeddings improved with text normalization, only GloVe 27B and 42B got statistically significant[7] increases.

Finally, in order to save time on training, we reduced the vocabulary of the word embedding lookup table to contain only words from the training data. This allowed us to reduce the network's training time by half and reduce its memory usage by over 90%. However, due to a bug, words outside of the train and dev set vocabulary were treated as unknown, considerably degrading our system's performance. When the vocabulary bug is fixed, our submission setting achieves performance with a retroactive rank of third place. See Table 6 for a comparison to other shared task entries taken from Strauss et al. (2016).

4.2 Lexicon Features

Usage of lexicons greatly improved performance, providing a statistically significant increase in f1-score for Collobert[8], GloVe 27B[9], GloVe 42B[10], and word2vec-400MT[11] embeddings (see Tables 7 and 8).

Figure 3 (left) shows a heat map of lexical coverage. As many of the cells along the diagonal are bright, it shows that we were able to produce lexicons for many categories with high coverage and low ambiguity. However, there are some notable exceptions, such as the `Location` lexicon showing high coverage on both `facility` and `geo-loc`, and both `Music` and `Person` lexicons showing high coverage on `musicartist`. This lexical overlap likely contributes to misclassification errors; the confusion matrix in Figure 3 (right) shows that misclassifications between `facility` and `geo-loc` and `musicartist` and `person` are quite frequent. Some lexical overlap makes sense considering the fact that sports teams will often include city names such as *Montreal Canadiens* or *Philadelphia Eagles*.

Table 8 compares our fixed shared task submission's entity-level f1-scores with and without lexicons. These results show that while many lexicons were effective – particularly `company`, `geo-loc`, and

[6] It is surprising that word2vec-400MT underperforms Glove 42B, despite its superior word type coverage, but this could be due to differences in training algorithm, preprocessing (word2vec-400MT used Ritter et al. (2011)'s Twitter NLP Tools), or casing (word2vec-400MT preserved case, while Glove 42B did not). We also evaluated 300 dim. GloVe embeddings trained on 840B words of Common Crawl data with a vocabulary size of 2.2M, however, they underperformed the GloVe 42B embeddings.

[7] Wilcoxon rank sum test, $p < 0.05$.

[8] Wilcoxon rank sum test, $p < 0.005$.

[9] Wilcoxon rank sum test, $p < 0.05$.

[10] Wilcoxon rank sum test, $p < 0.05$.

[11] Wilcoxon rank sum test, $p < 0.01$.

NE	com.	fac.	geo-loc	movie	musicartist	other	person	product	sportsteam	tvshow
− lex	38.94	29.00	61.16	3.43	28.68	26.54	55.75	8.48	35.34	10.77
+ lex	45.74	29.87	64.13	4.18	23.06	27.84	55.01	9.04	35.20	9.76
∆	+6.80	+0.87	+2.97	+0.75	−5.63	+1.30	−0.74	+0.57	−0.13	−1.01

Table 8: Per-category comparison of our fixed shared task submission settings with and without lexicons.

				B-sports				
(1)	Don't	be	biased	**Argentina**	destroyed	them	...	
				B-geo-loc				
	O	B-sports	I-sports					
(2)	**#SunDevils**	Sun	Devils	struggle	to	beat	FCS	...
	O	O	O					
	O		O		B-company			
(3)	#Fedex	,	#microsoft	,	#Twitter	...		
	O		O		O			

Figure 4: Examples of (1) contextual ambiguity (2) tagging inconsistencies and (3) hashtags inconsistencies. The upper tag is gold annotation. The lower tag is our system's prediction.

other – the lexicons MusicArtist, Person, SportsTeam, and TVShow were detrimental to NER performance. As noted above, the MusicArtist and Person lexicons had substantial overlap, most likely contributing to poor performance.

4.3 Error Analysis

In this section, we describe different sources of errors from a subsample of mistagged test set entities.

4.3.1 Unseen Entities

One of the biggest source of errors when trying to tag noisy Web-text is the amount of unseen entities the system will face. In the WNUT dataset, roughly 40% of the entities present in the test set are not in the train or dev datasets. This underscores the importance of high-coverage word embeddings, lexicon construction, and lexical matching, since the tagger has not encountered almost half of the entities.

4.3.2 Contextual Ambiguity

With fine-grained entities such as the ones defined for this task, our system tends to make errors due to confusion between entity classes. Figure 3 shows the confusion matrix when the system is evaluated over the test dataset. One common error occurs between geo-loc and other classes, more specifically company, facility and sportsteam. We extracted 50 examples for each type of confusion and found out that place names were mostly being tagged as geo-loc even though context indicates otherwise. Figure 4 shows a few examples.

Another important class ambiguity is between musicartist and person. In a subsample of 64 examples, 49 were tagged as person. Furthermore, the entity matched both entity's lexicons in 59% of the cases. This is also supported by the confusion matrix where music artists get tagged as person more than twice as often as they get tagged correctly. This contextual ambiguity seems to have led to a few tagging inconsistencies that could also explain lower overall performance. Either from train to test set or within the same set, entities sometimes ended up with multiple tags or no tags at all. Such examples are: singers like *Justin Bieber* being tagged as person in the training set and musicartist in the test set; devices such as *BlackBerry* being tagged either as company or product. Some of these inconsistencies are understandable because most of the time more than one tag could fit[12]. Refining lexicons to maximize coverage while minimizing ambiguity remains an essential area of future work.

4.3.3 Hashtags

In tweets, hashtags are omnipresent. They are a way to highlight relevant keywords or phrases making it easier to categorize the tweets they are in. It then becomes important to be able to retrieve important

[12] It could also be explained by the fact that the dataset consists of data constructed with different time periods and annotators.

information from those relevant keywords. From the subsample we observed that most entities containing a hashtag were not tagged at all. This can be explained by the fact that only 4% of hashtags are part of entities in the training set making our network biased against tagging hashtags. This likely lead to more errors on the test set where more than 15% of hashtags are part of entities.

5 Related Research

Named entity recognition is a task with a long history, dating back to MUC-7 (Chinchor and Robinson, 1997). In this section, we describe the NER research that influenced our system and give an overview of the work on NER for Twitter. For a more detailed survey, see (Chiu and Nichols, 2016).

Most recent approaches to NER have been characterized by the use of CRF, SVM, and perceptron models, where performance is heavily dependent on feature engineering. Ratinov and Roth (2009) used non-local features, a gazetteer extracted from Wikipedia, and Brown-cluster-like word representations. Lin and Wu (2009) used phrase features obtained by performing k-means clustering over a private database of search engine query logs in place of a lexicon. Passos et al. (2014) proposed a model that infused word embeddings with lexical knowledge. In order to combat the problem of sparse features, Suzuki et al. (2011) performed feature reduction with large-scale unlabelled data.

Recently, the state-of-the-art for NER neural networks have overtaken other approaches to NER. Most approaches build on the pioneering work of Collobert et al. (2011), which showed that word embeddings could be employed in a deep FFNN to achieve near state-of-the-art results on POS tagging, chunking, NER, and SRL. Santos et al. (2015) augmented the architecture of Collobert et al. (2011) with character-level CNNs, reporting improved performance on Spanish and Portuguese NER. Huang et al. (2015) employed BLSTMs in place of FFNNs for the POS-tagging, chunking, and NER tasks, but they employed heavy feature engineering instead of using a CNN to automatically extract character-level features. Lample et al. (2016) proposed LSTM-CRF and Stack-LSTM architectures for NER.

The earliest work on NER for Twitter, used a CRF model with global features from tweet clusters to conduct NER with the MUC-7 4 class task definition (Liu et al., 2011). Ritter et al. (2011) developed a suite of NLP tools explicitly for Twitter and expanded the task to the 10 class definition used in the WNUT shared tasks. A key difference between NER for Twitter and conventional NER is that the former also considers peripheral tasks such as named entity tokenization (Li et al., 2012), normalization (Liu et al., 2012), and linking (Guo et al., 2013; Yamada et al., 2015). The WNUT 2015 Shared Task included text normalization and named entity tokenization and detection tasks (Baldwin et al., 2015), with most systems using machine learning methods like CRF together with a variety of features including lexicons, orthographic features, and distributional information. In contrast with conventional NER, there was only one neural network entry (Godin et al., 2015), and most systems tended to prefer Brown clusters to word embeddings. The state of the art at WNUT 2015 used a cascaded model of entity tokenization, followed by linking to knowledge bases, and, finally, classification with random forests (Yamada et al., 2015).

Our system adopts the architecture of Chiu and Nichols (2016), which combined BLSTMs to maximize context over the tagged word sequence and word-level CNNs to automatically generate character-level features with a partial-matching lexicon to achieve the state-of-the-art for NER on both CoNLL 2003 and OntoNotes datasets. Our system can be viewed as an investigation into how well state-of-the-art neural approaches adapt to the challenges of NER on noisy Web data.

6 Conclusion

In this paper, we described the DeepNNNER entry to the WNUT 2016 Shared Task #2: Named Entity Recognition in Twitter, which adopted the BLSTM-CNN model of Chiu and Nichols (2016). Extensive evaluation showed that high word type coverage for word embeddings is crucial to NER performance, likely due to rare words in entities, and that both text normalization and partial matching on lexicons constructed from DBpedia (Auer et al., 2007) contribute significantly to performance. Our best-performing system uses text normalization, lexicon partial matching, and the GloVe word embeddings of Pennington et al. (2014) trained on 42B words of Common Crawl data, and it achieves a maximum F1-score of 47.24, performance surpassing that of the shared task's second-ranked system.

Acknowledgments

This research was supported by Honda Research Institute Japan Co., Ltd. We would like to thank Jason P.C. Chiu for the initial DeepNNNER implementation, the WNUT organizers for hosting this shared task, our anonymous reviewers for their useful feedback, the DBpedia contributors for the useful source of lexical knowledge, and Pennington et al. (2014) for releasing the GloVe word embeddings.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

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. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015)*, Beijing, China, July. Association for Computational Linguistics.

Colin Cherry, Hongyu Guo, and Chengbi Dai. 2015. Nrc: Infused phrase vectors for named entity recognition in twitter. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015)*, pages 54–60. Association for Computational Linguistics.

Nancy Chinchor and Patricia Robinson. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, page 29.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Marc Claesen, Jaak Simm, Dusan Popovic, Yves Moreau, and Bart De Moor. 2014. Easy hyperparameter search using Optunity. In *Proceedings of the International Workshop on Technical Computing for Machine Learning and Mathematical Engineering*.

Maurice Clerc and James Kennedy. 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73.

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

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 (WNUT 2015)*, pages 146–153, Beijing, China, July. Association for Computational Linguistics.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of The 12th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1020–1030.

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

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

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 721–730. ACM.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038. Association for Computational Linguistics.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 359–367. Association for Computational Linguistics.

Xiaohua Liu, Ming Zhou, Furu Wei, Zhongyang Fu, and Xiangyang Zhou. 2012. Joint inference of named entity recognition and normalization for tweets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 526–535. 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 *Proceedings of the Twenty-seventh Annual Conference on Advances in Neural Information Processing Systems*, pages 3111–3119.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 285–290. IEEE.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152. Association for Computational Linguistics.

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.

Cıcero Santos, Victor Guimaraes, RJ Niterói, and Rio de Janeiro. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entities Workshop*, pages 25–33.

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)*.

Jun Suzuki, Hideki Isozaki, and Masaaki Nagata. 2011. Learning condensed feature representations from large unsupervised data sets for supervised learning. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers*, pages 636–641. Association for Computational Linguistics.

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*, pages 142–147. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015)*, Beijing, China, July. Association for Computational Linguistics.