Comparing LSTM and FOFE-based Architectures for Named Entity Recognition

Klang, Marcus; Nugues, Pierre

2018

Comparing LSTM and FOFE-based Architectures for Named Entity Recognition

Marcus Klang, Pierre Nugues

Department of Computer Science
Lund University, S-221 00 Lund
Marcus.Klang@cs.lth.se, Pierre.Nugues@cs.lth.se

Abstract

LSTM architectures (Hochreiter and Schmidhuber, 1997) have become standard to recognize named entities (NER) in text (Lample et al., 2016; Chiu and Nichols, 2016). Nonetheless, Zhang et al. (2015) recently proposed an approach based on *fixed-size ordinally forgetting encoding* (FOFE) to translate variable-length contexts into fixed-length features. This encoding method can be used with feed-forward neural networks and, despite its simplicity, reach accuracy rates matching those of LTSMs in NER tasks (Xu et al., 2017). However, the figures reported in the NER articles are difficult to compare precisely as the experiments often use external resources such as gazetteers and corpora. In this paper, we describe an experimental setup, where we reimplemented the two core algorithms, to level the differences in initial conditions. This allowed us to measure more precisely the accuracy of both architectures and to report what we believe are unbiased results on English and Swedish datasets.

1. Introduction

Named entity recognition (NER) aims at identifying all the names of persons, organizations, geographic locations, as well as numeric expressions in a text. This is a relatively old task of NLP that has applications in multiples fields such as information extraction, knowledge extraction, product recommendation, and question answering. Named entity recognition is also usually the first step of named entity linking, where the mentions of named entities, once recognized, are disambiguated and linked to unique identifiers (Ji and Nothman, 2016; Ji et al., 2017).

Over the time, NER has used scores of techniques starting from hand-written rules, to decision trees, support vector machines, logistic regression, and now deep neural networks. The diversity of applications and datasets makes it difficult to compare the algorithms and systems. Researchers in the field quickly realized it and the committee of the message understanding conferences (MUC) first defined procedures for a systematic evaluation of NER performance (Grishman and Sundheim, 1996). The CoNLL 2002 and 2003 conferences (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) further developed them and provided standardized annotations, multilingual datasets, and evaluation scripts, that are still references today.

In spite of a continuous research, designing a perfect domain-independent NER is still an unmet goal. New ideas and architectures make that the state-of-the-art is improving every year. However, the figures reported in the NER articles are difficult to compare precisely as the experiments often involve external resources such as gazetteers and non-published corpora.

In this paper, we describe an experimental setup, where we reimplemented two of the best reported algorithms and where we defined identical initial conditions. This allowed us to measure more precisely the accuracy of both architectures and to report what we believe are unbiased results on English and Swedish datasets.

2. Previous Work

NER has been addressed by many techniques. Participants in the MUC conferences, such as FASTUS, used extensively gazetteers and regular expressions to extract the mentions (Appelt et al., 1993). The CoNLL conferences started to distribute annotated corpora that enabled participants to train classifiers such as logistic regression, decision trees, perceptrons, often organized as ensembles. For a review of early systems from 1991 to 2006, see Nadeau and Sekine (2007).

With the advent of deep learning, long short-term memory architectures (LSTM) (Hochreiter and Schmidhuber, 1997) have become standard to recognize named entities. Out of the 24 teams participating in the trilingual entity disambiguation and linking task (EDL) of TAC 2017, 7 used bidirectional LSTMs – with varying degrees of success (Ji et al., 2017).

Chiu and Nichols (2016) reported a score of 91.62 on the CoNLL 2003 test set with LSTM and convolutional neural networks (CNN) on character embeddings using the development set and the training set to build their model; Lample et al. (2016) used LSTM and conditional random fields (CRF) and reached 90.94 on the same test set; Ma and Hovy (2016) combined LSTM, CNN, and CRF and obtained 91.21.

Parallel to the LSTM achievements, Zhang et al. (2015) recently proposed an approach based on *fixed-size ordinally forgetting encoding* (FOFE) to translate variable-length contexts into fixed-length features. This encoding method can be used with feed-forward neural networks and, despite its simplicity, reach accuracy rates matching those of LTSMs in NER tasks (Xu et al., 2017).

All the reported performance figures are now close and may be subject to initialization conditions of random seeds. See Reimers and Gurevych (2017) for a discussion on their validity. In addition, all the experiments are carried out on the same data sets, again and again, which may, in the long run, entail some data leaks.

In this paper, we report experiments we have done with reimplementations of two of the most accurate NER taggers on English, to be sure we could reproduce the figures and that we applied to the Swedish Stockholm-Umeå corpus (SUC) (Ejerhed et al., 1992).

3. Datasets and Annotations

Annotated datasets. As datasets, we used the English corpus of CoNLL 2003, OntoNotes, and SUC, that bracket the named entities with semantic categories such as location, person, organization, etc. The corpora use either IOB v1 or v2 as annotation tagsets. We converted the annotation to IOBES, where S is for single-tag named entities, B, for begin, E, for end, I, for inside, and O for outside. For the bracketed example from CoNLL:

> Promising 10th-ranked [$_{MISC}$ American $_{MISC}$] [$_{PER}$ Chanda Rubin $_{PER}$] has pulled out of the [$_{MISC}$ U.S. Open Tennis Championships $_{MISC}$] with a wrist injury, tournament officials announced.

the annotation yields:

> Promising/O 10th-ranked/O American/S-MISC Chanda/B-PER Rubin/E-PER has/O pulled/O out/O of/O the/O U.S./B-MISC Open/I-MISC Tennis/I-MISC Championships/E-MISC with/O a/O wrist/O injury/O ,/O tournament/O officials/O announced/O ./O

The CoNLL 2003 dataset is derived from the Reuters corpus (RCV).

Word embeddings. For English, we used the pre-trained Glove 6B embeddings (Pennington et al., 2014) and the lower-cased 100 to 300 dimension variants. In addition, we trained our own cased and lowercased embeddings using the Word2vec algorithm provided by the Gensim library (Řehůřek and Sojka, 2010). For Swedish, we used Swectors (Fallgren et al., 2016) and we trained Swedish embeddings from the Swedish Culturomics Gigaword Corpus (Eide et al., 2016).

4. Systems

We implemented two systems: one based on FOFE, which is an extension to that of Klang et al. (2017) and Dib (2018) and the second one on LSTM, taking up the work of Chiu and Nichols (2016).

4.1 FOFE

The FOFE model can be seen as a weighted bag-of-words (BoW). Following the notation of Xu et al. (2017), given a vocabulary V, where each word is encoded with a one-hot encoded vector and $S = w_1, w_2, w_3, ..., w_n$, an arbitrary sequence of words, where e_n is the one-hot encoded vector of the nth word in S, the encoding of each partial sequence z_n is defined as:

$$z_n = \begin{cases} 0, & \text{if n = 0} \\ \alpha \cdot z_{n-1} + e_n, & \text{otherwise,} \end{cases} \quad (1)$$

where the α constant is a weight/forgetting factor which is picked such as $0 \leq \alpha < 1$. The result of the encoding is a vector of dimension $|V|$, whatever the size of the segment.

Features. The neural network uses both word and character-level features. The word features extend over parts of the sentence, while character features are only applied to the focus words: The candidates for a potential entity.

Word-level Features. The word-level features use bags of words to represent the focus words and FOFE to model the focus words as well as their left and right contexts. As context, we used all the surrounding words up to a maximum distance. The beginning and end of sentence are explicitly modeled with `BOS` and `EOS` tokens, which have been added to the vocabulary list.

Each word feature is used twice, both in raw text and normalized lower-case text. The FOFE features are used twice, both with and without the focus words. For the FOFE-encoded features, we used $\alpha = 0.5$. The complete list of features is then the following:

- Bag of words of the focus words;
- FOFE of the sentence: starting from the left, excluding the focus words; starting from the left, including the focus words; starting from the right, excluding the focus words; and starting from the right, including the focus words.

This means that, in total, the system input consists of 10 different feature vectors, where five are generated from the raw text, and five generated from the lowercase text.

Character-Level Features. The character-level features only model the focus words from left to right and right to left. We used two different types of character features: One that models each character and one that only models the first character of each word. We applied the FOFE encoding again as it enabled us to weight the characters and model their order. For these features, we used $\alpha = 0.8$. Higher choice of alpha for character features matches the original implementation. Our hypothesis is, using a higher alpha for the FOFE encoded character features increases its likelihood to remain salient during training.

Training. NER datasets are traditionally unbalanced with regards to the negative outside class. To produce enough positive examples to fit the model, we balanced every mini-batch, so that it contains a constant and adjustable ratio of positive and negative classes. The size of an epoch is defined by the number of mini-batches we can fill with the smallest class repeated T times.

4.2 LSTM

The LSTM model uses the sequential input directly, which does not require any preprocessing. We feed the network with the input sentences. Before training as a performance optimization, we sorted all the sentences by length and we then divided them into mini-batches. This reduces the amount of masking, and thereby wasteful computations as the majority of mini-batches will be of fixed length.

We use the same set of input features as Chiu and Nichols (2016):

- Word-level, the matching word-embedding for the input word or the unknown word embedding if the word is not in our vocabulary.
- Word-character level, all the characters per word are mapped to embeddings trained with the model. We extracted the alphabet manually and the language is specific.
- Word-case feature, per word class mapping such as lower, upper, title, digits etc.

Architecture. The word-character level features are passed through a convolution layer with a kernel of size 3 and a max-pooling layer with a window matching the maximum word length, resulting in a fixed-width character feature.

We tested LSTM cell sizes of dimension 100 and 200, our character embedding set at 30, and a maximum word length at 52. Dropout was set to 50% for recurrent LSTM connections, character feature and before the output layer. We observed that the output dropout had the greatest influence on the results.

All the word and character features are then concatenated per word and fed to a single BILSTM layer consisting internally of two independent LSTM cells which represent the forward and backward passes. The BILSTM output is the concatenation of both passes. We computed the tag scores for the BILSTM-CNN model using softmax from a single dense layer. The BILSTM-CNN-CRF model replaces the dense softmax layer with a CRF layer.

We used a negative log likelihood as loss function for the BILSTM-CNN-CRF model and categorical crossentropy for BILSTM-CNN.

5. Experimental Setup

We implemented all the models using Keras and Tensorflow as its backend. Early stopping was performed on all the models with a patience ranging from 5 to 10 depending on model; the parameters from the best epoch were selected for the resulting classifier. The word-embeddings were preinitialized without any preprocessing or normalization. In addition, we froze them during training but in a future work we may enable training. All the models used the Nadam optimizer.

Hyperparameters. We carried out a minimal hyperparameter search for BILSTM variants as usable parameters could be found in previous work. However, we could not use FOFE parameters as they produced poor results for us. We performed a smaller hyperparameter search on the CoNLL 2003 dataset to find more optimal parameters.

Evaluation. All the models produce IOBv2 annotations, IOBES is postprocessed by simple rules into correct IOBv2 tags. The annotated datasets were evaluated using conlleval from the CoNLL 2003 task, using tab delimiter instead of space, this because SUC3 has tokens with spaces in them.

SUC3 is evaluated on the 4 statistically significant classes instead of all 9: PERSON, PLACE, INST and MISC. The MISC is the combination of the remaining 5. Ontonotes 5 is evaluated on PERSON, GPE, ORG, NORP, LOC and MISC using the same principle as SUC. Following (Chiu and Nichols, 2016), we excluded the New Testaments portion from Ontonotes 5 as it lacks goldstandard annotations for NER.

For crossvalidation, we indexed all the sentences of the full dataset and we randomly split the index into 10 folds; this created 10 sets of indices. For each fold, we used one of them as test set and the rest as training set. For the training part, we used a 90/10% split to create a validation part which is used to determine when to stop training. Finally, we combined the predictions of the test part in each fold, 10 of them, into one dataset which we evaluated to produce the final score.

6. Results

BILSTM models outperform FOFE-CNN, as can be seen in Table 1. We trained FOFE-CNN models on Ontonotes 5 and SUC 3 with similar settings as the CoNLL 2003 dataset, these parameters produced subpar models which were not comparable without a new hyperparameter search.

Character features are important, as can be seen in Table 3 with more substantial improvements for lowercase embeddings. CRF improves the result for most embeddings and larger networks appear to have mixed results.

Acknowledgements

This research was supported by Vetenskapsrådet, the Swedish research council, under the *Det digitaliserade samhället* program.

References

Douglas Appelt, Jerry Hobbs, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. 1993. SRI: Description of the JV-FASTUS system used for MUC-5. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore*, pages 221–235, San Francisco, August. Morgan Kaufmann.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the ACL*, 4:357–370.

Firas Dib. 2018. A multilingual named entity recognition system based on fixed ordinally-forgetting encoding. Master's thesis, Lund University, Lund.

Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. 2016. The Swedish culturomics gigaword corpus: A one billion word Swedish reference dataset for NLP. In *From Digitization to Knowledge 2016: Resources and Methods for Semantic Processing of Digital Works/Texts, Proceedings of the Workshop*, volume 126, pages 8–12, Krakow.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project. Technical Report 33, Department of General Linguistics, University of Umeå.

Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of swedish word vectors. In *Sixth Swedish Language Technology Conference (SLTC), Umeå 17-18 nov 2016*.

CoNLL03 Test	Glove 6B 100d			Glove 6B 200d			Glove 6B 300d			RCV 256d		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	83.64	83.82	83.73	83.27	83.46	83.37	82.85	82.81	82.83	85.51	87.66	86.57
BILSTM-CNN 100d	86.87	90.00	88.41	85.39	88.77	87.05	84.75	88.74	86.70	86.25	89.09	87.65
BILSTM-CNN-CRF 100d	89.25	90.10	89.67	88.63	89.20	88.92	88.61	88.56	88.59	89.25	89.41	**89.33**
BILSTM-CNN-CRF 200d	89.76	90.44	**90.10**	89.08	89.82	**89.45**	88.80	88.83	**88.81**	88.92	88.92	89.07
FOFE-CNN	79.87	84.17	81.97	82.26	83.11	82.68	83.64	83.16	83.40	87.91	87.80	87.86

Table 1: CoNLL03 Test results

Ontonotes5 Test	Glove 6B 100d			Glove 6B 200d			Glove 6B 300d		
	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	81.32	82.46	81.89	82.65	80.03	81.32	81.49	80.09	80.79
BILSTM-CNN 100d	84.20	86.91	85.53	82.59	85.84	84.19	82.92	86.40	**84.62**
BILSTM-CNN-CRF 100d	82.70	86.50	84.56	83.91	84.10	84.00	83.13	84.84	83.97
BILSTM-CNN-CRF 200d	84.62	86.79	**85.69**	86.43	86.00	**86.22**	81.86	84.50	83.16

Table 2: Ontonotes 5 Test results.

SUC3 Test	Swectors 300d			Gigawords 256d			Gigawords Lcase 256d		
	P	R	F1	P	R	F1	P	R	F1
BILSTM 100d	81.82	64.07	71.87	81.91	77.98	79.90	79.14	75.99	77.53
BILSTM-CNN 100d	79.72	74.83	77.20	82.33	81.79	82.06	82.23	80.46	81.34
BILSTM-CNN-CRF 100d	82.55	78.31	80.37	86.13	83.28	**84.68**	82.53	82.12	**82.32**
BILSTM-CNN-CRF 200d	85.09	77.48	**81.11**	81.63	79.47	80.54	82.15	80.79	81.47
SUC3 10-fold crossvalidation									
BILSTM-CNN-CRF 100d	82.07	80.22	81.14	85.22	83.62	84.41	84.31	84.32	84.31

Table 3: SUC 3 Test results

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471.

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

Heng Ji and Joel Nothman. 2016. Overview of TAC-KBP2016 Tri-lingual EDL and Its Impact on End-to-End KBP. In *Proceedings of the Ninth Text Analysis Conference (TAC 2016)*, Gaithersburg.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, and Cash Costello. 2017. Overview of TAC-KBP2017 13 Languages Entity Discovery and Linking. In *Proceedings of the Tenth Text Analysis Conference (TAC 2017)*, Gaithersburg.

Marcus Klang, Firas Dib, and Pierre Nugues. 2017. Overview of the ugglan entity discovery and linking system. In *Proceedings of the Tenth Text Analysis Conference (TAC 2017)*, Gaithersburg, Maryland, November.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1)*, pages 1064–1074.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on EMNLP*, pages 1532–1543, Doha.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on EMNLP*, pages 338–348, Copenhagen.

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 CoNLL-2003*, pages 142–147, Edmonton.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158, Taipei.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the ACL (Volume 1)*, pages 1237–1247, Vancouver.

ShiLiang Zhang, Hui Jiang, MingBin Xu, JunFeng Hou, and LiRong Dai. 2015. The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP (Volume 2)*, pages 495–500.