A Multi-task Learning Approach for Named Entity Recognition using Local Detection

Nargiza Nosirova, Mingbin Xu, Hui Jiang
Department of Electrical Engineering and Computer Science
Lassonde School of Engineering, York University
4700 Keele Street, Toronto, Ontario, Canada
{nana, xmb, hj}@cse.yorku.ca

Abstract

Named entity recognition (NER) systems that perform well require task-related and manually annotated datasets. However, they are expensive to develop, and are thus limited in size. As there already exists a large number of NER datasets that share a certain degree of relationship but differ in content, it is important to explore the question of whether such datasets can be combined as a simple method for improving NER performance. To investigate this, we developed a novel locally detecting multitask model using FFNNs. The model relies on encoding variable-length sequences of words into theoretically lossless and unique fixed-size representations. We applied this method to several well-known NER tasks and compared the results of our model to baseline models as well as other published results. As a result, we observed competitive performance in nearly all of the tasks.

1 Introduction

Named entity recognition aims to solve the problem of detecting proper nouns in a text and categorizing them into different types of entities. This information is useful for higher-level NLP applications such as summarization and question answering (Aramaki et al., 2009; Ravichandran and Hovy, 2002). NER systems have been originally built by applying hand-crafted features and other external resources to achieve good results (Ratinov and Roth, 2009). In the recent years, researchers have turned to neural network architectures. For example, Collobert et al. (2011) introduced a neural network model that learns important features from word embeddings, thus requiring little feature engineering. However, in his use of FFNNs, the context used around a word is restricted to a fixed-size window. This bears the risk of losing potentially relevant information between words that are far apart. Recently, Xu et al. (2017) proposed a local detection approach for NER by making use of a technique that can encode any variable-length sequence of words into a theoretically lossless and unique fixed-size representation. This technique, called the Fixed-size ordinally forgetting encoding (FOFE), has the ability to capture immediate dependencies within the sentence, and thus using the encodings as features partly overcomes the limitations of FFNNs. Using FOFE features practically eliminate any need for feature engineering. Furthermore, it is known that FFNNs are universal approximators, and its advantages over RNNs include easier tuning, faster training times, and a simpler implementation. Therefore, since the main drawbacks of FFNNs are resolved by FOFE features, they are acceptable for recognition.

Meanwhile, learning many associated tasks in parallel has been shown to improve performance compared to learning each task separately (Bakker and Heskes, 2003; Caruana, 1997). One of the more popular MTL approaches is hard-parameter sharing, which has the advantage of reducing the chance of over-fitting (Baxter, 1997) while also being simple to implement. Generally, MTL is applied by using auxiliary tasks that are similar to the main task. For example, Martínez Alonso and Plank (2017) uses auxiliary tasks such as Part-of-speech (POS) and chunking for main tasks such as NER.

Our main contribution lies in combining these two proposed models for the NER task. We investigate how hard parameter sharing can be used to improve NER models, while also further exploring the idea of using auxiliary NER tasks to boost the performance of a main NER task. In this paper, we propose a novel multi-task FOFE-based FFNN model with the aim of generalizing the underlying distributions of the named entities in the data. We report our experimental results on several popular NER tasks. Our method has yielded competitive results and improved performance in comparison

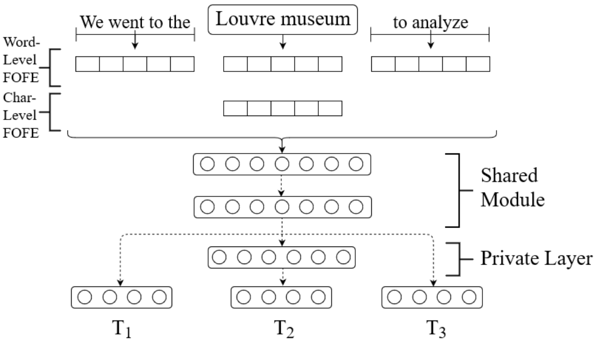

Figure 1: Illustration of an example network structure for our MTL model using FOFE codes. The window currently examines the fragment *Louvre Museum*.

to the baselines in all tasks.

This paper is organized as follows. Section 2 introduces the FOFE technique, while Section 3 outlines how we use FOFE to extract features for the NER task. Section 4 gives an overview of our model, including the MTL technique used. From Section 5 onwards, we outline the experimental setup, present our results and state our conclusion.

2 Fixed-Size Ordinally Forgetting Encoding (FOFE)

Consider a vocabulary V, where each word can be represented by a 1-of-$|V|$ one-hot vector. Let $S = w_1 \cdots w_N$ denote a sequence of N words from V, and denote $\boldsymbol{e_n}$ to be the one-hot vector of the n-th word in S, where $1 \leq n \leq N$. Assuming $\boldsymbol{z_0} = \mathbf{0}$, the FOFE code $\boldsymbol{z_n}$ of the sequence from word w_1 to w_n is as follows:

$$\boldsymbol{z_n} = \alpha \cdot \boldsymbol{z_{n-1}} + \boldsymbol{e_n}$$

where α is a constant forgetting factor. Hence, $\boldsymbol{z_n}$ can be viewed as a fixed-size representation of the subsequence $\{w_1, w_2, \cdots, w_n\}$. Following the theoretical properties presented by Zhang et al. (2015) in Appendix A, we see that FOFE can uniquely and losslessly encode any sequence of variable length into a fixed-size representation.

3 Extracting Features using FOFE

Word-level Features We extract the bag-of-words of the focus token as well as the FOFE encoding of its left and right contexts. All of the word features are computed in both case sensitive and case insensitive forms. The FOFE encodings are further projected to lower-dimensional dense vectors using projection matrices for both the case sensitive and insensitive forms. Those matrices are initialized using word embeddings pre-trained with *word2vec* (Mikolov et al., 2013), and are tuned during training.

Character-level Features Based on a pre-defined set of all possible characters, we view the focus token as a case-sensitive character sequence and encode it using FOFE from left to right, as well as right-to-left. We then project the character encodings using a trainable character embedding matrix. For a fair comparison, we also use character CNNs to generate additional character-level features (Kim et al., 2015).

4 A MTL approach for NER using FOFE

Consider k learning tasks $\{T_i\}_{i=1}^{k}$, where each task T_i is associated with an input-output pair of sequences $(x_{1:n}, y^i_{1:n})$, where $x_j \in W$ and $y^i_j \in Y_i$. The input set W is shared by all tasks, whereas the output sets Y_i are reserved to a single corresponding task. At each training step, we randomly choose a task T_i and training sample $(x_{1:n}, y^i_{1:n}) \in T_i$. We forward pass the training sample through the shared layers to predict the labels $\hat{y}^i_j$, calculate the loss based on the true labels y^i_j and backpropagate for parameter update. The training sample of task T_i is eventually fed into its corresponding task-specific softmax layer for classification, whereas the hidden layers are shared by all tasks. Additionally, we may attribute additional private hidden layers to T_i, located between the shared layer and softmax layer. If T_i is a main task, the private layers can be useful for personalizing the learning of the task, since some of information contained in the training signals distinct to the task may be swamped in the shared layers by the auxiliary tasks' training signals. If T_i is an auxiliary task, it would enable the shared layers to focus on representing information pertinent to the other tasks, while keeping its distinct signals in its private layer. This is especially useful if T_i has a large data size compared to the main task.

Figure 1 represents an instance of the model. The character and word features extracted using FOFE are concatenated to form the input to the model. Each of the hidden layers are fully-connected. The model has many outputs, corresponding to the number of tasks trained for the specific instance.

5 Experiments

We perform experiments for multiple NER tasks in English (ENG), Spanish (SPA) and Chinese (CMN). When performing MTL, the auxiliary tasks are from the same language as the main task. Both the main and auxiliary tasks are trained for NER. For each main task, we consider the following systems: (1) Baseline model, trained without any auxiliary task. (2) A system involving a combination of auxiliary tasks along with the main task.

Main and auxiliary tasks We experiment with five main tasks: (1) The CoNLL-03 English task and CoNLL-02 Spanish task, annotated with 4 entity types. We abide by the standard split defined in (Tjong Kim Sang and De Meulder, 2003; Tjong Kim Sang, 2002). (2) The OntoNotes English and Chinese dataset, tagged with 18 entity types. We follow the standard split described in (Pradhan et al., 2013). (3) The KBP 2016 (Ji and Nothman, 2016) tri-lingual task (English, Spanish and Chinese), tagged with 5 named entity types and 5 nominal entity types. Since this task only comes with a test set, we use the training set of KBP 2015 as well as an in-house dataset labelled in a similar fashion, The in-house dataset consists of 10k English and Chinese documents. The results for this task are generated using the official evaluator.

For each main task, we use the remaining tasks from the same language as auxiliary tasks. For example, if CoNLL-03 English is the main task, we use the OntoNotes and KBP 2016 English tasks as auxiliary tasks. Detailed info about the tasks can be found in Appendix A.

Baselines We use the model of Xu et al. (2017) as our baseline. We apply the implementation[1] released by the author to train the models with our tasks.

Training and hyperparameters We use cross entropy as our objective function. Training is executed by using mini-batch SGD with momentum of 0.9. Learning rates are exponentially decayed by a factor of 1/16 if dev-performance drops compared to the last run. We apply dropout to all layers with a value of 0.5. We set all the forgetting factors for words to $\alpha_w = 0.5$, and characters to $\alpha_c = 0.8$. All the hidden layers consist of ReLUs, and are initialized based on a uniform distribution following Glorot et al. (2011). We follow the same post-processing and decoding steps for named entities as the ones outlined in Xu et al. (2017). We performed grid search and selected the hyper-parameters over the main task's development set, with early stopping. We implemented the neural network using the Tensorflow library. For detailed hyper-parameter settings, please go to Appendix A.

LANG	FOFE-FFNN **F1 (%)**	MTL-FOFE-FFNN **F1 (%)**	Liu et al. (2016) **F1 (%)**
ENG	0.750	**0.770**	**0.772**
CMN	0.698	0.717	0.762
SPA	0.700	**0.722**	0.736
ALL	**0.718**	**0.738**	**0.756**

Table 1: KBP 2016 tri-lingual results for our MTL model with the baseline and best model. All of the available auxiliary tasks are used for each model.

6 Results and Discussion

Tables 1 to 5 present the results of our proposed model on the benchmark tasks. The results are compared to our baseline model as well as other published results, including the state-of-the-arts. We refer to our baseline model (Xu et al., 2017) as FOFE-FFNN. Our proposed model is referred to as MTL-FOFE-FFNN, and we specify the used auxiliary tasks in brackets.

Overall, the multi-task models yield better performance over baselines for nearly all of the tasks. The task that has benefited the most from MTL is the KBP 2016 trilingual task, whose results are summarized in Table 1. The results are broken down according to the three languages. Our proposed model outperforms the baseline by 2.0 F1 points overall, with the most increase coming from

Model	**F1 (%)**
Collobert et al. (2011)	89.59
Huang et al. (2015)	90.10
Strubell et al. (2017)	90.54
Yang et al. (2016)	**90.94**
Luo et al. (2015)	91.2
Lample et al. (2016)	90.94
Chiu and Nichols (2016)	**91.62**
FOFE-FFNN	90.71
MTL-FOFE-FFNN (+ KBP 2016 ENG)	90.85
MTL-FOFE-FFNN (+ OntoNotes ENG)	90.78
MTL-FOFE-FFNN (+ OntoNotes ENG + KBP 2016 ENG)	90.91

Table 2: English NER results (CoNLL-2003) on the test set. The three sections, in order, are models: trained with training set only, trained with both training and dev set, our baseline and model.

[1] https://github.com/xmb-cipher/fofe-ner

Model	F1 (%)
Strubell et al. (2017)	**86.84**
Chiu and Nichols (2016)	**86.28**
Durrett and Klein (2014)	84.04
FOFE-FFNN	85.88
MTL-FOFE-FFNN (+ CoNLL-03 ENG)	85.91
MTL-FOFE-FFNN (+ KBP 2016 ENG)	86.03
MTL-FOFE-FFNN (+ CoNLL-03 ENG + KBP 2016 ENG)	**86.06**

Table 3: English NER results (OntoNotes) on the test set.

Model	F1 (%)
Che et al. (2013)	69.82
Pappu et al. (2017)	67.2
FOFE-FFNN	72.96
MTL-FOFE-FFNN (+ KBP 2016 ZH)	**73.32**

Table 4: Chinese NER results (OntoNotes) on the test set.

the Spanish model, with a boost of 2.2 F1 points. Compared to Liu et al. (2016), our English model comes closest to within 0.2 F1 points. The system by Liu et al. (2016) is a combination of two different models, and each is based on 5-fold cross-validation.

In Table 2, we compare our model for CoNLL-2003 with baseline and previously published results. We train our proposed model on a combination of auxiliary tasks, and observe that using all of the auxiliary tasks reaches the best performance. Compared to the models that are trained without the dev-set, our highest result only comes second to Yang et al. (2016). We notice that the KBP 2016 auxiliary task yields slightly better results than OntoNotes. This may be due to the OntoNotes dataset having a lot more source variety. CoNLL-03 consists mainly of newswire data. KBP 2016 is similarly built from news articles, with the addition of discussion forum posts.

The OntoNotes English and Chinese results are presented in Tables 3 and 4, and the CoNLL-2002 Spanish results in Table 5. In Table 3, we observe respectable gains over the baseline models, and are within 0.2 F1 points from Chiu and Nichols (2016), who make use of bi-LSTM CNNs. We should also mention that we do not use any hand-crafted features. For the Chinese OntoNotes task, we found two non-neural network method results and have exceeded them with both our baseline and proposed models. We have been unable to find prior published neural-based results, and thus believe (although with uncertainty) that we have established state-of-the-art results.

Model	F1 (%)
dos Santos and Guimarães (2015)	82.21
Gillick et al. (2016)	82.95
Lample et al. (2016)	**85.75**
Yang et al. (2016)	**85.77**
FOFE-FFNN	83.22
MTL-FOFE-FFNN (+ KBP 2016 SPA)	**84.11**

Table 5: Spanish NER results (CoNLL-2002) on the test set.

7 Related Work

Named entity recognition Recently, methods involving deep learning have been very successful in many NLP projects. Due to the limitations of FFNNs, more powerful neural networks, such as recurrent neural networks (RNNs), have been used. Many studies have used bidirectional Long Short-Term memory (B-LSTM) architectures along with CRFs (Luo et al., 2015; Huang et al., 2015), and have reported strong NER results. As for character-level modelling, studies have turned to convolutional neural networks (CNNs). For instance, dos Santos and Guimarães (2015) have employed CNNs to extract character-level features for Spanish and Portuguese, and obtained successful results.

Multi-task learning Much of the work done in MTL has been initiated by Caruana (1997). His techniques have been used and confirmed in many studies (Maurer et al., 2016; Ando and Zhang, 2005). The success of MTL has been associated with label entropy, regularizers, training size and many other aspects (Martínez Alonso and Plank, 2017; Bingel and Søgaard, 2017). For example, Collobert and Weston (2008) use MTL in a unified model to train multiple core NLP tasks: NER, Part-of-Speech, chunking and semantic role labeling with neural networks. They show that MTL improves generality among the shared tasks. Liu et al. (2015) used MTL for information retrieval and semantic classification by training a model for both tasks which has shared and private layers. Their method exceeded performance of strong baselines for tasks such as query classification and web search.

8 Conclusion

In this paper, we explored the advantages of multi-task learning using FOFE as a possible solution for improving performance on various NER tasks, as well as the benefit of having auxiliary NER tasks. We applied this method to several well-known NER tasks and observed competitive results, without using any external resources or hand-crafted features.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. 2009. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185–192. Association for Computational Linguistics.

Bart Bakker and Tom Heskes. 2003. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99.

Jonathan Baxter. 1997. A bayesian/information theoretic model of learning to learn viamultiple task sampling. *Mach. Learn.*, 28(1):7–39.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 52–62, Atlanta, Georgia. Association for Computational Linguistics.

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

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.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California. Association for Computational Linguistics.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. PMLR.

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

Heng Ji and Joel Nothman. 2016. Overview of tackbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. In *Proceedings of the Nineth Text Analysis Conference (TAC2016)*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-aware neural language models. *CoRR*, abs/1508.06615.

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. Association for Computational Linguistics.

Dan Liu, Wei Lin, Shiliang Zhang, Si Wei, and Hui Jiang. 2016. Neural networks models for entity discovery and linking. *CoRR*, abs/1611.03558.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*, pages 912–921. The 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. Association for Computational Linguistics.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 17(1):2853–2884.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Aasish Pappu, Roi Blanco, Yashar Mehdad, Amanda Stent, and Kapil Thadani. 2017. Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 365–374, New York, NY, USA. ACM.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Bjrkelund, 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, Sofia, Bulgaria. 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*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 41–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cícero Nogueira dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. *CoRR*, abs/1505.05008.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2670–2680.

Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA. 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 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247, Vancouver, Canada. Association for Computational Linguistics.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *CoRR*, abs/1603.06270.

Shiliang Zhang, Hui Jiang, Mingbin Xu, Junfeng Hou, and Li-Rong Dai. 2015. A fixed-size encoding method for variable-length sequences with its application to neural network language models. *CoRR*, abs/1505.01504.