AlphaBrains at WojoodNER shared task: Arabic Named Entity Recognition by Using Character-based Context-Sensitive Word Representations

Toqeer Ehsan[1], Amjad Ali[2], Ala Al-Fuqaha[2]
[1]Department of Computer Science, University of Gujrat, Gujrat, Pakistan
[2]Information and Computing Technology (ICT) Division, College of Science and Engineering (CSE), Hamad Bin Khalifa University, Doha, Qatar
toqeer.ehsan@uog.edu.pk, amsali@hbku.edu.qa, aalfuqaha@hbku.edu.qa

Abstract

This paper presents Arabic named entity recognition models by employing single-task and multi-task learning paradigms. The models were developed by using character-based contextualized Embeddings from Language Model (ELMo) in the input layers of the Bidirectional Long-Short Term Memory (BiLSTM) networks. The ELMo embeddings are quite capable of learning the morphology and contextual information of tokens in word sequences. The single-task learning model outperformed the multi-task learning model, achieving micro F_1-scores of 0.8751 and 0.8884, respectively, ranking 10^{th} and 7^{th} in the shared task for flat and nested NER.

1 Introduction

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task which aims at identifying and extracting sub-sequences of the text associated with Named Entities (NEs). These NEs are subsequently categorized into different semantic groups, such as names, places, organizations, events and dates, etc. NER is considered a crucial preliminary task for the development of different applications, such as, information retrieval (Popovski et al., 2020), text summarization (Khademi and Fakhredanesh, 2020), machine translation (Vu et al., 2020), topic modeling and event discovery (Feng et al., 2018), word-sense disambiguation (Al-Hajj and Jarrar, 2022) and others. NER is a typical sequence labeling token classification task where each token is assigned a tag. IOB labeling is a common method employed for annotating datasets for NER.

Different machine and deep learning techniques have been used to perform NER, such as, Conditional Random Fields (CRF) (Patil et al., 2020; Bhumireddypalli et al., 2023), Support Vector Machines (SVM) (Mady et al., 2022), template-based (Cui et al., 2021), Recurrent Neural Networks (RNN) (Ahmad et al., 2020), Bidirectional LSTM (Tehseen et al., 2023), Transformer-based Models (e.g. BERT) (Jarrar et al., 2022; Agrawal et al., 2022) and others. On the other hands, the nested NER has also been performed by employing LSTM with CRF inference (Dadas and Protasiewicz, 2020), LSTM-based hierarchical layering model along with contextual word representations (Wang et al., 2020), bidirectional LSTMs with exhaustive representations (Sohrab and Miwa, 2018), BERT embeddings based LSTM-CRF (Straková et al., 2019), fine-tuning pre-trained BERT model (Jarrar et al., 2022) and others.

This paper presents model development and results of a shared task for Arabic NER (Jarrar et al., 2023). The shared task has been divided into two sub-tasks, flat NER[1] and nested NER[2]. The flat NER uses a conventional annotation scheme, however, the nested scheme provides a hierarchical annotation within the NEs. For the shared task, a different version of the Wojood dataset (Jarrar et al., 2022) has been used which has 70% data for training, 10% for development and 20% for evaluation purposes. The nested NEs are challenging to predict as multiple output layers are required to train. However, the nested annotation provides a deeper insight of overlapping NEs.

We developed two models which are based on single and multi-task learning. Both models are based on long-short term memory networks. Furthermore, transfer learning has been used to enhance the models' learning capability. The contextualized pre-trained ELMo embeddings have been incorporated with word embeddings at the input layers of the models. The ELMo embeddings significantly enhanced the results as compared to the Word2Vec and part of speech(POS) tagging. The POS tags were used as encoding vectors which were concatenated with token encoding vectors.

[1]https://codalab.lisn.upsaclay.fr/competitions/11740
[2]https://codalab.lisn.upsaclay.fr/competitions/11750

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 783–788

Both single and multi-task learning models used *softmax* non-linearity for multi-class token classification. The details of the proposed models are discussed in the Section 3. The single task learning model performed better than the multi-task learning model and produced competitive results as compared to the baseline provided in the shared task. Rest of the paper describes the dataset, proposed single task and multi-task learning NER models, results and conclusion.

2 Data

The shared task released a version of the dataset from Jarrar et al. (2022). The training and development sets have IOB labels whereas the test set has been released without labels for evaluation purposes. Table 1 shows the label-wise distribution of NEs for training and development sets. Table 2 further presents the sentence and token distribution among all three sets.

3 System

We developed neural models by using Bidirectional Long-Short Term Memory (BiLSTM) networks. The BiLSTM model has the ability to learn context within token sequences for the token classification tasks (e.g. named entity recognition). A bidirectional model has two LSTM layers, the first layer reads the tokens in the forward direction whereas the second layer scans the tokens in the backward direction. The two way scanning is helpful to attain the contextual information within the token sequences. The input sequence of N words x_1, x_2,..., x_n is given as the input. Equation 1 shows the BiLSTM($x_{1:n}$,i) function which demonstrates union of the forward and backward layers.

$$BiLSTM(x_{1:n}, i) = LSTM_f(x_{1:i}) \circ LSTM_r(x_{n:i}) \quad (1)$$

The function shows the representation to a vector *i* by conditioning the previous context $x_{1:i}$ and the forthcoming sequence $x_{n:i}$. The models are based on two implementation paradigms; i) Single Task Learning (STL) and ii) Multi-Task Learning (MTL).

3.1 The Proposed Single Task Learning Model

The proposed STL-based model is comprised of word encodings, word embeddings, pre-trained word representations, BiLSTM-based hidden layers, and a single output layer. Figure 1 shows the architecture of our proposed STL model. The training and development samples have been converted to word encodings which are concatenated with embedding vectors at the input layer. The input layer contains embedding layers along with pre-trained ELMo embeddings vectors. Both embedding vectors are concatenated and fed to the hidden BiLSTM layers. The hidden layers produce contextual representations which are used to perform multi-class classification by employing *softmax* non-linearity function as shown in Equation 2.

$$o_i = Softmax(Xh_i + b) \quad (2)$$

Where o_i represents the output for ith instance, h_i shows hidden state of ith instance in the sequence along with the weights X and the bias b. The model has a single output layer to produce one label for each input token. The STL model has been trained for both flat and nested NER. The flat NEs are trained just like a standard sequence labeling task. However, for nested NER, we combined the NE labels with a delimiter to make it a single label. Section 4 presents the results of STL model for flat and nested labeling.

We experimented with three hidden BiLSTM layers. A *Dropout* layer is added after each hidden layer. The *keras* library has been used with *Tensorflow* back-end in Python-3 for the implementation of both models. The dimensions of the internal embeddings are set to 256 whereas the pre-trained ELMo embeddings have 1024 projection dimensions. Section 3.3 further describes the ELMo embeddings and transfer learning. Each hidden LSTM layer has 256 units with a dropout value of 0.2(20%). Root Mean Squared Propagation (RMSprop) optimizer has been used with a learning rate of 0.001. The loss function was the *categorical cross-entropy* for all the experiments. The sequence length has been set to have 256 tokens for each sentence. The models are trained for 15 epochs with a batch size of 128 samples. All the models have been trained using GPU servers available at the Scientific Compute Cluster (SCCKN)[3].

3.2 The Proposed Multi-task Learning Model

For the nested NER, a single entity can be annotated to have multiple layers of tags. Therefore, the multi-task learning is a suitable method. The MTL models hold a prominent position in the realm of research for conducting various NLP tasks including

[3]https://www.scc.uni-konstanz.de

IOB label	Train set			Dev set		
	$Count_{Flat}$	$Count_{Nested}$	$Total$	$Count_{Flat}$	$Count_{Nested}$	$Total$
CARDINAL	1,245	18	1,263	182	1	183
CURR	19	160	179	1	20	21
DATE	10,667	623	11,290	1,567	89	1,656
EVENT	1,863	71	1,934	253	14	267
FAC	689	191	880	85	26	111
GPE	8,133	7,167	15,300	1,132	1,031	2,163
LANGUAGE	131	1	132	15	0	15
LAW	374	0	374	44	0	44
LOC	510	109	619	63	13	76
MONEY	171	0	171	20	0	20
NORP	3,505	242	3,747	488	32	520
OCC	3,774	113	3,887	544	7	551
ORDINAL	2,805	683	3,488	410	94	504
ORG	10,731	2,444	13,175	1,566	303	1,869
PERCENT	105	0	105	13	0	13
PERS	4,496	498	4,994	650	80	730
PRODUCT	36	0	36	5	0	5
QUANTITY	44	2	46	3	0	3
TIME	286	2	288	55	0	55
UNIT	7	41	48	0	3	3
WEBSITE	434	0	434	45	0	45
Total	**50,025**	**12,365**	**62,390**	**7,141**	**1,713**	**8,854**

Table 1: Entity-wise statistics of train and development sets.

Category	No. of Sentences	No. of Tokens
Train set	16,817	394,499
Dev set	3,133	55,826
Test set	5,989	111,951
Total	**25,939**	**562,276**

Table 2: Number of sentences and tokens in train, development and test sets.

NER (Jarrar et al., 2022; Yan et al., 2023; Du et al., 2022; Fang et al., 2023). Figure 2 shows the architecture of the proposed MTL model. The proposed MTL model has 21 output layers associated with each NE label. The *softmax* non-linearity function is used for each output layer. The *softmax* function performs multi-class classification to predict an NE label or 'O' label. The MTL model has been trained for both flat and nested NER. The model performed better for the nested dataset because a single token may have multiple NE labels due to the nested nature of the text. We further performed MTL for flat NER by converting the flat dataset into 21 columns. The outputs from multiple output layers were then combined into a single label for each token. However, for flat NEs, it is challenging to find a single most appropriate label because the MTL model can predict multiple labels for a single token. The model setup and hyper-parameters are similar to the STL model.

3.3 Transfer Learning

Deep learning based models require larger datasets to produce state-of-the-art results. Mostly, the annotation of large datasets is not feasible. Therefore, the transfer learning is a suitable approach by training word embeddings on huge unannotated datasets. We have used ELMo embeddings which have been pre-trained on a large Arabic textual data (Che et al., 2018; Fares et al., 2017)[4]. Context-free word embeddings (Pennington et al., 2014; Mikolov et al., 2013; Bojanowski et al., 2017) provide a single word vector for each token irrespective of the context. However, contextual ELMo word embeddings (Peters et al., 2018) generate the vectors with respect to the character-based contextual information in a sentence. The ELMo model contains three neural network layers. First character-based convo-

[4]https://github.com/HIT-SCIR/ELMoForManyLangs

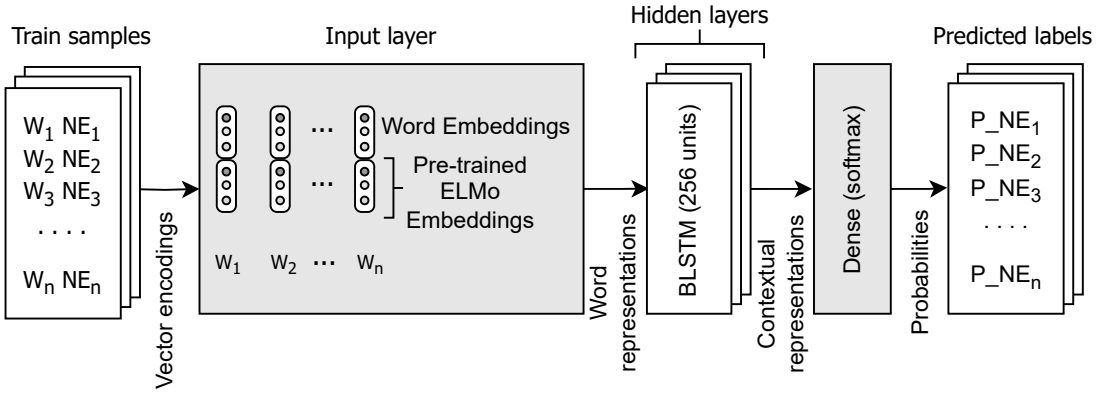

Figure 1: Architecture of the single task learning-based model.

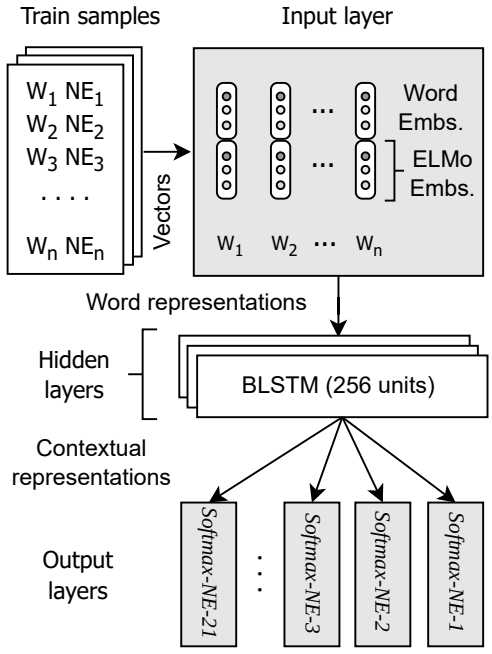

Figure 2: Architecture of multi-task learning-based model.

lutional layer, the second and the third layers are bi-directional LSTM networks to learn the contextual representations. Due to the character convolutions, the ELMo embeddings are quite capable to produce vectors for Out-Of-Vocabulary words. Both STL and MTL models have been trained by incorporating the ELMo vectors achieved from the third layer of the model showing significant improvements in the NER results.

4 Results and Discussion

The micro-F_1 score has been computed for the evaluation of the models by using *seqeval* Python package[5]. The results for the flat and nested NER on the 20% test set are shown in Tables 3 and 4.

Models	Pre.	Rec.	F_1
Baseline	–	–	0.8681
Our STL model	0.8745	0.8758	0.8751
Our MTL model	0.8647	0.8806	0.8726

Table 3: Flat NER results (micro F_1-score).

Table 3 shows the comparison of the proposed single and multi-task learning models with the baseline score for flat NER. Our proposed STL model performed better than the MTL and the baseline. However, there is a subtle difference in our models due to the nested nature of the dataset as a single token can have multiple IOB labels. The MTL model may produce multiple labels for flat NER against a single token therefore, for the selection of a single label, a naive approach has been used which selects the left-most label among multiple NE labels.

Models	Pre.	Rec.	F_1
Baseline	–	–	0.9047
Jarrar et al. (2022)	0.8772	0.8909	0.8840
Our STL model	0.8845	0.8923	0.8884
Our MTL model	0.8900	0.8793	0.8846

Table 4: Nested NER results (micro F_1-score).

Table 4 shows the results for nested NER from the proposed STL and MTL models and compares with the baseline and the F_1-score from Jarrar et al. (2022). While our results fall short of the baseline model, which is a transformer-based model, they outperform Jarrar et al. (2022). The STL model performs better than the MTL model for the nested

[5]https://pypi.org/project/seqeval

NER. For the nested NER to be trained on the STL model, we combined the labels by using a delimiter ($\sim$) and trained the dataset like flat labels. This label combination resulted in a total of 298 distinct labels. Beside the contextualized word embeddings, we also experimented by incorporating part of speech(POS) tags and Word2Vec embeddings. POS tagging has not shown any improvements for NER (Tehseen et al., 2022, 2023) and the F_1-score remained around $\sim$0.78. We used the Stanford POS tagger (Toutanova et al., 2003) to tag the Wojood NER dataset and concatenated the POS encoding vectors with the word encoding vectors at the input layers of the models. The Arabic Word2Vec (Soliman et al., 2017) improved the results but the F_1-scores still remained under 0.82. The ELMo emeddings showed significant improvements by producing competitive results for Arabic NER.

5 Conclusion

This paper presents the description of the models and their performances for two shared tasks; i) flat NER and ii) nested NER for Arabic. We proposed Bidirectional LSTM-based single and multi-task learning models for both types of datasets. The incorporation of character-based contextualized word embeddings produced competitive results as compared to the baseline provided in the shared task.

References

Ankit Agrawal, Sarsij Tripathi, Manu Vardhan, Vikas Sihag, Gaurav Choudhary, and Nicola Dragoni. 2022. BERT-based transfer-learning approach for nested named-entity recognition using joint labeling. *Applied Sciences*, 12(3):976.

Muhammad Tayyab Ahmad, Muhammad Kamran Malik, Khurram Shahzad, Faisal Aslam, Asif Iqbal, Zubair Nawaz, and Faisal Bukhari. 2020. Named entity recognition and classification for Punjabi Shahmukhi. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–13.

Moustafa Al-Hajj and Mustafa Jarrar. 2022. ArabGlossBert: Fine-tuning BERT on Context-Gloss Pairs for WSD. *arXiv preprint arXiv:2205.09685*.

Veera Sekhar Reddy Bhumireddypalli, Srinivas Rao Koppula, and Neeraja Koppula. 2023. Enhanced conditional random field-long short-term memory for name entity recognition in English texts. *Concurrency and Computation: Practice and Experience*, 35(9):e7640.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based Named Entity Recognition using BART. *arXiv preprint arXiv:2106.01760*.

Sławomir Dadas and Jarosław Protasiewicz. 2020. A bidirectional iterative algorithm for nested named entity recognition. *IEEE Access*, 8:135091–135102.

Xiaojing Du, Yuxiang Jia, and Hongying Zan. 2022. MRC-based Medical NER with Multi-task Learning and Multi-strategies. In *China National Conference on Chinese Computational Linguistics*, pages 149–162. Springer.

Qin Fang, Yane Li, Hailin Feng, and Yaoping Ruan. 2023. Chinese Named Entity Recognition Model Based on Multi-Task Learning. *Applied Sciences*, 13(8):4770.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. A language-independent neural network for event detection. *Science China Information Sciences*, 61:1–12.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER 2023: The First Arabic Named Entity Recognition Shared Task. In *Proceedings of the First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested Arabic Named Entity Corpus and Recognition using BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636.

Mohammad Ebrahim Khademi and Mohammad Fakhredanesh. 2020. Persian Automatic Text Summarization based on Named Entity Recognition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pages 1–12.

Lobna Ahmed Mady, Yasmine A Afify, and Nagwa Badr. 2022. Nested Biomedical Named Entity Recognition. *International Journal of Intelligent Computing and Information Sciences*, 22(1):98–107.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Nita Patil, Ajay Patil, and BV Pawar. 2020. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188.

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 (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *arXiv preprint arXiv:1802.05365*.

Gorjan Popovski, Barbara Koroušić Seljak, and Tome Eftimov. 2020. A survey of named-entity recognition methods for food information extraction. *IEEE Access*, 8:31586–31594.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2843–2849.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. *Procedia Computer Science*, 117:256–265.

Jana Straková, Milan Straka, and Jan Hajič. 2019. Neural architectures for nested NER through linearization. *arXiv preprint arXiv:1908.06926*.

Amina Tehseen, Toqeer Ehsan, Hannan Bin Liaqat, Amjad Ali, and Ala Al-Fuqaha. 2022. Neural POS Tagging of Shahmukhi by Using Contextualized Word Representations. *Journal of King Saud University-Computer and Information Sciences*.

Amina Tehseen, Toqeer Ehsan, Hannan Bin Liaqat, Xiangjie Kong, Amjad Ali, and Ala Al-Fuqaha. 2023. Shahmukhi Named Entity Recognition by using Contextualized Word Embeddings. *Expert Systems with Applications*, 229:120489.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich Part-of-Speech tagging with a cyclic dependency network. In *Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics*, pages 252–259.

Van-Hai Vu, Quang-Phuoc Nguyen, Kiem-Hieu Nguyen, Joon-Choul Shin, and Cheol-Young Ock. 2020. Korean-vietnamese neural machine translation with named entity recognition and part-of-speech tags. *IEICE Transactions on Information and Systems*, 103(4):866–873.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928.

Yibo Yan, Peng Zhu, Dawei Cheng, Fangzhou Yang, and Yifeng Luo. 2023. Adversarial Multi-Task Learning for Efficient Chinese Named Entity Recognition. *ACM Transactions on Asian and Low-Resource Language Information Processing*.