BiLSTM-CRF for Persian Named-Entity Recognition

ArmanPersoNERCorpus: the First Entity-Annotated Persian Dataset

Hanieh Poostchi[a,b], Ehsan Zare Borzeshi[b], Massimo Piccardi[a]
[a] University of Technology Sydney, [b] Capital Markets CRC
PO Box 123 Broadway NSW 2007, Australia; 55 Harrington Street The Rocks NSW 2000, Australia
{hpoostchi,ezborzeshi}@cmcrc.com, massimo.piccardi@uts.edu.au

Abstract

Named-entity recognition (NER) can still be regarded as work in progress for a number of Asian languages due to the scarcity of annotated corpora. For this reason, with this paper we publicly release an entity-annotated Persian dataset and we present a performing approach for Persian NER based on a deep learning architecture. In addition to the entity-annotated dataset, we release a number of word embeddings (including GloVe, skip-gram, CBOW and Hellinger PCA) trained on a sizable collation of Persian text. The combination of the deep learning architecture (a BiLSTM-CRF) and the pre-trained word embeddings has allowed us to achieve a 77.45% CoNLL $F1$ score, a result that is more than 12 percentage points higher than the best previous result and interesting in absolute terms.

Keywords: Named-entity recognition, recurrent neural networks, BiLSTM-CRF, Persian language, low-resource languages.

1. Introduction

Named-entity recognition (NER) is a natural language processing component that aims to identify all the "named entities" (NEs) such as names of people, locations, organisations and numerical expressions in an unstructured text. This information can be useful in its own right or facilitate higher-level NLP tasks such as text summarization and machine translation. To date, NER research has mostly focussed on languages with a high number of digitally annotated resources such as English and German (Tjong Kim Sang and De Meulder, 2003) and Spanish and Dutch (Tjong Kim Sang, 2002). The main reason why many other languages, including many from the Asian region, have received less attention is the significant scarcity of public, annotated digital resources. Amongst those, the Persian language is spoken by more than 110 million speakers world-wide and has more than 570K articles on Wikipedia. However, it has been rarely studied for NER (Khormuji and Bazrafkan, 2014) or even just text processing (Shamsfard, 2011).

Although language-agnostic NER systems such as Polyglot-NER (Al-Rfou et al., 2015) exist, their performance is generally not competitive in comparison to language-specific NER. For this reason, in our previous work (Poostchi et al., 2016) we developed a dedicated NER system for Persian[1]. Its development was supported by two datasets: a) a sizable unannotated dataset of Persian sentences for training word embeddings, and b) an entity-annotated dataset for training named-entity classifiers.

This paper makes three distinct contributions: 1) it officially releases the entity-annotated dataset with an ISLRN[2] that should make its utilisation easier; 2) it releases four different word embeddings trained on the unannotated resources for a comprehensive Persian dictionary of nearly 50K unique words, also available via an ISLRN[3]; 3) it proposes a deep learning Persian NER based on a state-of-the-art architecture, the BiLSTM-CRF (Huang et al., 2015; Lample et al., 2016). Thanks to this architecture and the trained word embeddings, we have been able to achieve an improvement of over 12 percentage points of CoNLL $F1$ score over our previous approach based on structural SVM (Poostchi et al., 2016).

2. Supervised and Unsupervised Datasets for Persian NER

Supervised NER usually involves two main steps: the unsupervised training of a word embedding from a large corpus, and the classification of named entities using an annotated dataset. This section describes the two datasets that we provide for NER in the Persian language.

2.1. The Unannotated Persian Corpus

An effective co-occurrence matrix can be calculated from an adequately large corpus of documents covering a range of contexts. To this end, we have collated three resources of Persian text: i) the *Leipzig corpora* (Goldhahn et al., 2012) with approximately 1M and 300K sentences from news websites and Wikipedia, respectively, ii) a subset of *VOA news* with 227K sentences[4], and iii) the *Persian Dependency Treebank* (Rasooli et al., 2013) with nearly 30K sentences.

The aggregated corpus, with a total number of sentences in excess of 1.6M, has gone through a pipeline of text normalisation and tokenisation (Feely et al., 2014) tools including PrePer (Seraji, 2013), the Farsi verb tokenizer

[1] Particularly, Western/Iranian Persian which is also known as Farsi.

[2] ISLRN: 399-379-640-828-6

[3] ISLRN: 921-509-141-609-6

[4] `http://www.ling.ohio-state.edu/~jonsafari/corpora/index.html#persian`

(Manshadi, 2013), SetPer (Seraji et al., 2012) and tok-tok (Dehdari, 2015). We refer the reader to (Poostchi et al., 2016) for more details.

After normalisation, we have trained four different word embeddings using the provided corpus. The methods are GloVe (Pennington et al., 2014), word2vec (both skip-grams and continuous bag of words) (Mikolov et al., 2013), and Hellinger PCA (HPCA) (Lebret and Collobert, 2014); they will be briefly explained in Section 3.1.. For every method, a window of size 5 in both directions was used to calculate the co-occurrence matrix. Then, only words with a minimum frequency of 15 were selected, resulting in a dictionary of $49,902$ distinct words. The length of the embedding vectors was set to 300. All these hyper-parameters were chosen empirically during an initial evaluation.

The collated corpus cannot be publicly released due to licensing restrictions on some of its parts. However, we have released all four word embeddings on GitHub[5], which will allow easy replication of our experiments.

2.2. The Entity-Annotated Persian Dataset

To create a Persian named-entity dataset, we have selected a subset of 7,682 news sentences from the **BijanKhan** (Bijankhan et al., 2011) corpus, which is the most-established POS tagged Persian corpus. The histogram of the sentences' length is shown in figure 1. The mode is around 24 words per sentence, but with a significant tail of longer sentences. An experienced annotator has led the annotation task and prepared a comprehensive instruction manual based on the definition of Sekine's extended named entities (Sekine, 2007). The annotation of the dataset was split over two native-speaking post-graduate students and disambiguated according to the context. For instance, word *Ferdowsi* has different labels in *"Ferdowsi*$_{B\text{-}ORG}$ *University*$_{I\text{-}ORG}$*"* and *"Ferdowsi*$_{B\text{-}PER}$*, the great epic poet"*. To evaluate the accuracy of the annotation, three other independent native-speaking reviewers have verified i) a random sample of 500 annotated NEs, and ii) a sample of 500 annotated NEs from the two most semantically-close classes (i.e., location and organisation). The percentages of corrections have only been 1.8% and 1.9%, respectively.

The annotated dataset, called **ArmanPersoNERCorpus**, contains a total of 250,015 tokens with a hit rate of 87.68% in the trained dictionary. Only 11.08% of the tokens are marked as part of entities (Poostchi et al., 2016). The NEs are annotated in IOB format and categorised into six pre-defined classes: *person*, *organisation*, *location*, *facility*, *product*, and *event*. More than 60% of the sentences have at least one entity of any type. Figure 2 shows the percentage of sentences containing at least one entity and a maximum between 1 and 7 entities of each particular class. The most frequent NE class is *organisation* with an appearance rate of more than 33% of the sentences. This is followed by *person* and *location* with more than 25%. *Event* and *product* are far less frequent with just over 6% and *facility* has the lowest frequency with about 4%.

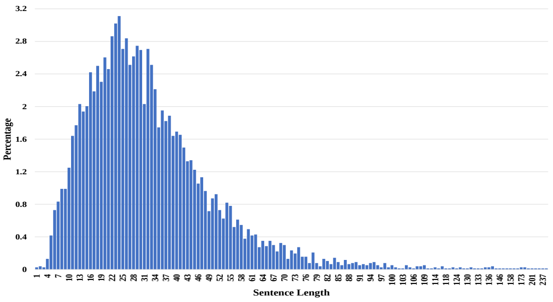

Figure 1: Histogram of the sentences' length in ArmanPersoNERCorpus.

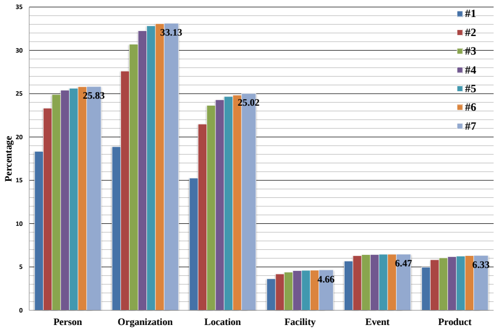

Figure 2: Percentage of sentences containing at least one entity and a maximum between 1 and 7 entities (in left-to-right order in the plot) of each particular named-entity class.

The dataset has been submitted to LR-MAP for global unique identification by an ISLRN. It is stored on GitHub and organised in the same 3 folds that we have used for the experiments. In addition to NER training, it could find use as an evaluation dataset for NER systems trained on silver standards.

3. Methods

Supervised NER is split into an initial step of word embedding followed by a step of token-level classification of the named entities. In this section we briefly describe the methods employed.

3.1. Word Embedding

A word embedding maps distinct words to high-dimensional feature vectors. GloVe (Pennington et al., 2014), word2vec (Mikolov et al., 2013), and Hellinger

[5] `https://github.com/HaniehP/PersianNER`

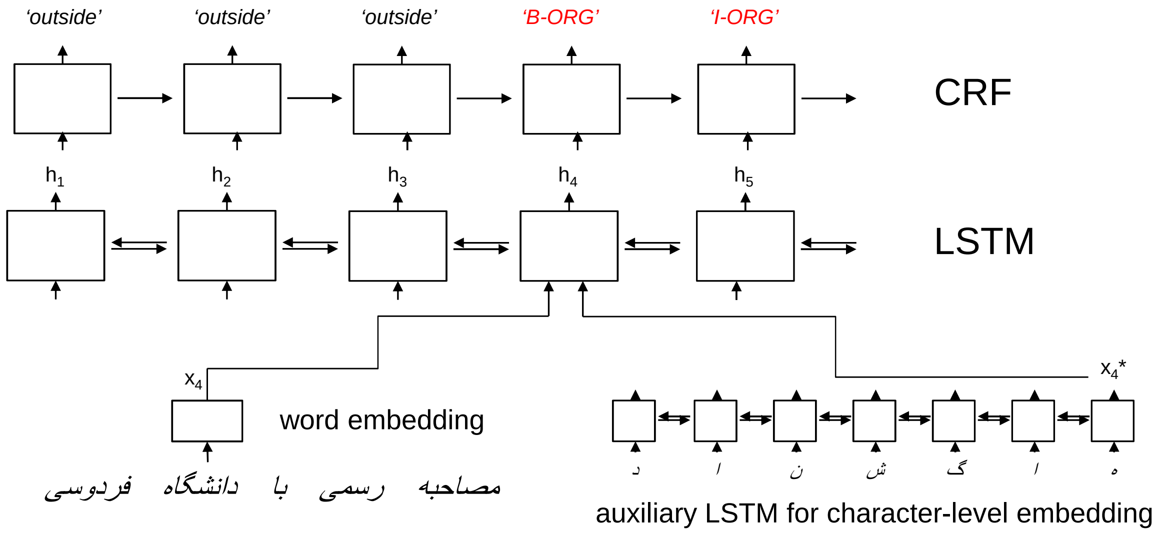

Figure 3: A diagram of the BiLSTM-CRF with an example of prediction. The input is a Persian sentence that consists of 5 tokens and translates into English as "an official interview with Ferdowsi University". The sentence is displayed in right-to-left order since this is how it would appear in Persian writing. However, this is not important for processing since both the tokens and their characters are processed in both directions. Token "University" is the 4-th token and its word embedding is noted as x_4 in the diagram. Its character-level embedding is the last output of the auxiliary LSTM and is noted as x_4^*. These embeddings are concatenated and used as the input of the corresponding slot in the main LSTM. In turn, the output of the LSTM slot is noted as h_4 and used as input for the CRF. Eventually, the CRF slot emits prediction "B-ORG". Token "Ferdowsi" is the 5-th token in the sentence and is predicted as "I-ORG".

PCA (HPCA) (Lebret and Collobert, 2014) are well-known examples of unsupervised word embeddings used successfully for NER.

GloVe is a global log-bilinear regression model with a weighted least-square objective that combines advantages of global matrix factorization and local context windows. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence.

Word2vec is a generative model for continuous representations of words that preserves the linear regularities amongst words. This model has two variants described hereafter: 1) the *skip-gram* model aims to learn word vector representations that are useful for predicting the nearby words in a sentence. A shallow neural network consisting of an input projection layer, an output layer and a softmax activation is trained to maximize the average of the log probability of a context word surrounding a given word; 2) the *continuous bag of words (CBOW)* model is similar to the skip-gram except that the roles of the input and output are reversed: in this model, the probability of the current word given the context is explicitly estimated.

HPCA is a simple spectral method analogous to PCA. First, the co-occurrence matrix is normalised row-by-row to represent the words by proper discrete probability distributions. Then, the resulting matrix is transformed into Hellinger space before applying PCA to reduce its dimensionality.

3.2. The BiLSTM-CRF for Sequential Labelling

The BiLSTM-CRF is a recurrent neural network obtained from the combination of a long short-term memory (LSTM) and a conditional random field (CRF) (Huang et al., 2015; Lample et al., 2016). The LSTM is used first to process each sentence token-by-token and produce an intermediate representation. Then, this intermediate representation is used as input for the CRF to provide the prediction of all the tokens' labels. The two models enjoy complementary features: as a complex, nonlinear model, the LSTM is able to effectively capture the sequential relationships amongst the input tokens; at its turn, the CRF permits optimal, joint prediction of all the labels in the sentence, capturing the relationships at label level. The "bi" in the name stands for "bidirectional" and alludes to the fact that the LSTM processes each sentence in both left-to-right and right-to-left order to embed the sequential dependencies in both directions. Before being processed, each token needs to be converted to a high-dimensional numerical vector, and this embedding is learned automatically alongside all the other parameters as part of the training stage. Eventually, the network also includes a second, auxiliary LSTM that further encodes each token character-by-character to capture the regularities at character level. Prior to being processed, also the individual characters need to be mapped to numerical embeddings. Figure 3 shows a complete diagram of the BiLSTM-CRF with an ample caption describing all the

	Entities						
Methods	**Person**	**Organization**	**Location**	**Facility**	**Event**	**Product**	**Overall**
CRF (HPCA)	64.10	42.25	57.97	41.09	22.48	20.00	49.92
Jordan-RNN (HPCA)	72.13	57.28	62.70	51.92	39.79	42.08	60.52
SVM-HMM (HPCA)	75.65	61.59	66.67	61.20	52.58	41.37	65.13
BiLSTM-CRF (HPCA)	77.69	69.70	69.67	57.33	52.69	49.24	69.43
BiLSTM-CRF (CBOW)	87.32	74.84	76.06	66.38	56.93	55.06	76.19
BiLSTM-CRF (GloVe)	86.97	75.73	76.62	67.41	55.58	55.08	76.58
BiLSTM-CRF (Skip-Gram)	**88.18**	**76.03**	**76.94**	**70.47**	**60.12**	**55.69**	**77.45**

Table 1: Comparison of Persian NER results with different classifiers and word embeddings (by class and as overall micro-average). The results above the double horizontal line are from (Poostchi et al., 2016) and are based on the same data and splits.

main variables and components (the character embeddings have been omitted to avoid cluttering).
Given a training set of labelled sequences, $\{x_i, y_i\}, i = 1 \dots N$, where x denotes a sequence of tokens and y the sequence of their labels, the BiLSTM-CRF is trained by maximizing the conditional log-likelihood:

$$\bar{w} = \operatorname*{argmax}_{w} \sum_{i=1}^{N} \ln p(y_i|x_i, w) \tag{1}$$

where w denotes all the model's parameters including the transition weights of the CRF, the weights of the main and auxiliary LSTMs, and the token and character embeddings. Once the model is trained, inference for a new sentence x is obtained as:

$$\bar{y} = \operatorname*{argmax}_{y} p(y|x, \bar{w}) \tag{2}$$

by propagating x through the network and applying the Viterbi algorithm at the CRF output layer.

4. Experimental Results

In this section, we present the NER results obtained with the BiLSTM-CRF and the different word embedding and we compare them with those reported in (Poostchi et al., 2016). For the experiments, we have used a TensorFlow implementation of the BiLSTM-CRF [6] (Dernoncourt et al., 2017), running each training session for 80 epochs (a value where the validation accuracy always seemed to have stabilised). For processing, all digits have been replaced with 0s. All hyper-parameters have been left to their default values.

Table 1 shows a comparison of the CoNLL $F1$ scores (by class and as overall micro-average) over the NER task for the various classifiers. The CoNLL $F1$ score is a strict version of the standard $F1$ score where a true positive is scored only if all the tokens of a given named entity are classified correctly (including their B- and I- tags). Conversely, every incorrect B- prediction is counted as a false positive. All the experiments have been performed with three-fold cross validation, using each of the three folds in turn as the test set and the other two for training. Moreover, each experiment has been repeated three times to mollify the effects of the random initialisation of the network's weights. This means that the values reported in Table 1 for our system are the average of $3 \times 3 = 9$ runs.

As shown in Table 1, the scores achieved by the BiLSTM-CRF have been higher than any of the results previously presented in (Poostchi et al., 2016). The results with the different word embeddings have ranged from a minimum average of 69.43% $F1$ score with HPCA to a maximum of 77.45% with the skip-gram. This relative ranking seems in good accordance with other NER results from the literature (Huang et al., 2015; Ma and Hovy, 2016). Amongst the classes, *person* is clearly the easiest and *product* the most challenging. This could be explained by the fact that the latter has much fewer samples (6% vs 25% of sentences) or that its patterns are possibly more diverse and harder to learn. In all cases, the proposed system has managed to outperform the best previous results for all classes and by a remarkable 12.32 $F1$ score percentage points on average.

5. Conclusion

In this paper, we have presented an approach for Persian NER based on a deep learning architecture and released a Persian annotated corpus alongside four different Persian word embeddings based on GloVe, CBOW, skip-gram and HPCA. The proposed approach has achieved an average $F1$ score of 77.45% which, to the best of our knowledge, is the highest Persian NER $F1$ score reported in the literature by 12.32 percentage points over the previous best result. Moreover, in addition to NER, the released word embeddings could find future use in other Persian NLP tasks including translation, question answering and summarisation.

6. Acknowledgements

This research has been funded by the Capital Markets Cooperative Research Centre (CMCRC).

[6] https://github.com/Franck-Dernoncourt/NeuroNER

7. Bibliographical References

Al-Rfou, R., Kulkarni, V., Perozzi, B., and Skiena, S. (2015). Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of 2015 SIAM International Conference on Data Mining*.

Dehdari, J. (2015). A fast, simple, multilingual tokenizer. `https://github.com/jonsafari/tok-tok/`.

Dernoncourt, F., Lee, J. Y., and Szolovits, P. (2017). NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

Feely, W., Manshadi, M., Frederking, R. E., and Levin, L. S. (2014). The cmu metal farsi nlp approach. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4052–4055, Reykjavik, Iceland.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Khormuji, M. K. and Bazrafkan, M. (2014). Persian named entity recognition based with local filters. *International Journal of Computer Applications*, 100(4):1–6.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Lebret, C. D., and Collobert, R. (2016). Neural architectures for named entity recognition. In *The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HCT-NAACL)*, pages 260–270, San Diego, California, USA.

Lebret, R. and Collobert, R. (2014). Word embedding through hellinger pca. In *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 482–490, Gothenburg, Sweden.

Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Manshadi, M. (2013). Farsi verb tokenizer. `https://github.com/mehdi-manshadi/Farsi-Verb-Tokenizer/`.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, Nevada, USA.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Poostchi, H., Borzeshi, E. Z., Abdous, M., and Piccardi, M. (2016). Personer: Persian named-entity recognition. In *The 26'th International Conference on Computational Linguistics (COLING)*, pages 3381–3389, Osaka, Japan.

Sekine, S. (2007). The definition of sekine's extended named entities. `http://nlp.cs.nyu.edu/ene/version7_1_0Beng.html`. New York University.

Seraji, M., BeÃ¡ta, M., and Joakim, N. (2012). A basic language resource kit for persian. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2245–2252, Istanbul, Turkey.

Seraji, M. (2013). Preper: A pre-processor for persian. In *Proceedings of Fifth International Conference on Iranian Linguistics*, Bamberg, Germany.

Shamsfard, M. (2011). Challenges and open problems in persian text processing. *Proceedings of LTC*, 11.

Tjong Kim Sang, E. F. and De Meulder, F. (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.

Tjong Kim Sang, E. F. (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.

8. Language Resource References

Bijankhan, M., Sheykhzadegan, J., Bahrani, M., and Ghayoomi, M. (2011). Lessons from building a persian written corpus: Peykare. *Language resources and evaluation*, 45(2):143–164.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *The 8th International Language Ressources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey.

Rasooli, M. S., Kouhestani, M., and Moloodi, A. (2013). Development of a persian syntactic dependency treebank. In *The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HCT-NAACL)*, pages 306–314, Atlanta, USA.