EasyChair Preprint

№ 4224

PUNER - Parsi ULMFiT for Named-Entity Recognition in Persian Texts

Fazlourrahman Balouchzahi and H. L. Shashirekha

September 21, 2020

PUNER-Parsi ULMFiT for Named-Entity Recognition in Persian Texts

F. Balouchzahi[1]and H. L. Shashirekha[2]

Department of Computer Science, Mangalore University, Mangalore - 574199, India
[1]frs_b@yahoo.com, [2]hlsrekha@gmail.com

Abstract. Named Entity Recognition (NER) is an information extraction technique for the automatic recognition of named entities and their classification in a natural language text. Applications of NER include extracting named entities from texts such as academic, news and medical documents, content classification for news providers, improving the search algorithms, etc. Most of the NER research works explored is for high resource languages such as English, German, and Spanish. Very less NER related work is done in low-resource languages such as Persian, Indian, and Vietnamese due to lack of annotated corpora for these languages. Among the mentioned languages very few works have been reported for the Persian language NER till now. Hence, this paper presents PUNER – a Persian NER system using Transfer Learning (TL) model that makes use of Universal Language Model Fine-tuning (UMLFiT) for NER in Persian language. This is accomplished by training a Language Model on Persian wiki text and using that model to develop a system for identifying and extracting named entities from the given Persian texts. Performance of the proposed model is compared with the Deep Learning (DL) models using BiLSTM by applying five word embedding models namely, Fasttext, HPCA, Skipgram, Glove, and COBOW and conventional Machine Learning (ML) model. All the models are evaluated on two Persian NER datasets and the results illustrate that TL model performs better than ML and DL models.

Keywords: NLP, Named Entity Recognition, Persian language, Transfer Learning, Machine Learning, ULMFiT, Deep Learning, Bidirectional LSTM.

1 Introduction

A Named Entity (NE) is the noun representing the name of a person, place, organization, etc., in general, newswire domain and Named Entity Recognition (NER) is the process of recognizing NEs in a given text and classifying them into one of the predefined categories [1]. It is an important preprocessing technique for many applications such as event detection from news, customer support for on-line shopping, knowledge graph construction, Information Retrieval, Question-Answering (QA) [2]. The information gained from NER task is useful in its own right and it also facilitates higher-level Natural Language Processing (NLP) tasks such as text summarization and machine translation [3].

Research in NER has mainly focused on high-resource languages such as English, German, and Spanish which have a large number of digitally annotated resources [4]. However, due to scarcity or inaccessibility of large volume of annotated digital resources and the challenges associated with the languages, NER for Persian language has received very less attention [4]. Hence, this paper presents PUNER – a Persian NER system using Transfer Learning (TL) model that makes use of Universal Language Model Fine-Tuning (ULMFiT) for NER in Persian language.

The rest of the paper is organized as follows: Section 2 gives an overview of the work carried out in the related area. TL, Machine Learning (ML), and Deep Learning (DL) models are explained in Section 3. The proposed methodology is discussed in Section 4 followed by Experiments and Results in Section 5. Section 6 gives the conclusion and throws light on future work.

2 Related Work

Researchers have developed many tools and techniques for identifying and classifying NEs for many languages and in several domains including the popular newswire. S. Amarappa et al., [1], has proposed a Supervised Multinomial Naïve Bayes NER model for Kannada Corpus consisting of parts of EMILLE (Enabling Minority Language Engineering) corpus and articles collected from internet and Kannada language books. The model trained on a corpus consisting of around 95170 words recognizes the NEs with an average F-measure of 81% and 10 fold cross-validation F-measure of 77.2%. An approach for Persian NER based on DL architecture using BiLSTM-CRF has been presented by H Poostchi et al., [3]. They have also released *ArmanPersoNERCorpus*, an Entity-Annotated Persian dataset and four different Persian Word Embedding (WE) models based on GloVe, CBOW, skip-gram, and HPCA. Their approach has achieved an average F1 score of 77.45% using Skip-Gram WE which is the highest Persian NER F1 score reported in the literature. A variety of Long Short-Term Memory (LSTM) based models for sequence tagging are proposed by Huang, Z et al., [5]. They have compared the performance of LSTM networks, bidirectional LSTM (BI-LSTM) networks, LSTM with a Conditional Random Field (CRF) layer (LSTM-CRF) and bidirectional LSTM with a CRF layer (BI-LSTM-CRF) for sequence tagging. Their models were tested on three NLP tagging tasks: Penn Treebank (PTB) POS tagging, CoNLL 2000 chunking, and CoNLL2003 NE tagging.

A web-based text classification application built by Indra S T et al., [6], classifies tweets into four predefined groups such as health, music, sport, technology using Logistic Regression. The classified tweets are presented according to the selected topics in an efficient format, such as a graph or table which also displays the accuracy of text classification. Khormuji, M.K et al. [4], presents NER in Persian Language using Local Filters model and publically available dictionaries to recognize Persian language NEs. Their NER framework is composed of two stages: detection of NE candidates using dictionaries for lookups and then filtering them based on false positives. They have reported 88.95% precision, 79.65% recall, and 82.73% F1 score. N

Taghizadeh et al., [7], have proposed a model forNSURL-2019 Task 7 which focuses on NER in Farsi. The objective of this task was to compare different approaches to find phrases that specify NEs in Farsi texts, and to establish a standard test bed for future researches on this task in Farsi. The best performance of 85.4% F1 score was obtained by MorphoBERT system [8] based on the phrase-level evaluation of seven classes of NEs including person, organization, location, date, time, money, and percent. They used morphological features of Farsi words together with the BERT model and Bi-LSTM.G S Mahalakshmi et al., [9], have proposed an NER system by applying Naïve Bayes algorithm for NER which takes Tamil text about temples as input. The system identifies the NEs in temple domain after preprocessing and parsing the dataset of 5000 documents collected from 'www.temples.dinamalar.com' and gives an accuracy of 79%.

A text classification model based on ULMFiT which is effective, extremely simple and efficient TL method is proposed by Jeremy H et al., [10]. They have also proposed several novel fine-tuning techniques that prevent catastrophic forgetting and enable robust learning across a diverse range of tasks. Their model is applied on three common text classification tasks, namely, sentiment analysis, question classification, and topic classification and all results are reported in terms of error rates. Sentiment analysis is evaluated on the binary movie review IMDb dataset and the binary and five-class version of the Yelp review dataset. Question Classification model is evaluated on the six-class version of the small TREC dataset, dataset of open-domain and fact-based questions divided into broad semantic categories. Topic classification is evaluated on the large-scale AG news and DBpedia ontology datasets.

3 Learning Models

NER system consists of two steps: recognizing the NEs and classifying them into one of the predefined set of labels or tags. While the first step is typically a segmentation problem where nouns or names are defined to be contiguous spans of tokens, with no nesting, the second phase is classification task that automatically assigns a suitable label/tag to a name from a predefined set of labels/tags [11]. NER can also be considered as a sequence labeling problem as every term/phrase in a sequence will be assigned a label from a predefined set of labels including ‘other’ as the label for non-NEs [12].

Researchers have explored several learning models for the task of text classification as well as sequence labeling problems. Off late TL is gaining importance as a learning model and showing better performance compared to conventional ML models and DL models.

3.1 Machine Learning model

Conventionally, in ML NER system, sentences are extracted from raw texts and then tokenization into words followed by tagging every word with one of the predefined NER tags. This tagged data in the form of <word, tag> is used to build the NER

model which is then used to predict the label of the input words/phrases. Large annotated data is required to build a ML model. Once the model is trained, it is tested on the test set and the accuracy is computed.

Instead of using a single ML model for text classification, it is advantageous to use an ensemble of learning models just like considering the decision of a team rather than an individual. The weakness of one learning model may be overcome by the strength of the other model in ensemble learning. One such ensemble learning model is Voting Classifier which is made up of 'n' classifiers/learning models. All these classifiers accept the same input simultaneously and predict the appropriate tag for each input. Then based on majority voting the tag with higher number of votes will be assigned to the input. Fig. 1 shows the structure of the voting classifier.

The major drawback of the ML approach is the availability of large annotated data and the tedious feature extraction process to obtain a good performance. Further, the strong dependence on domain knowledge for designing features makes the method difficult to easily generalize to new tasks.

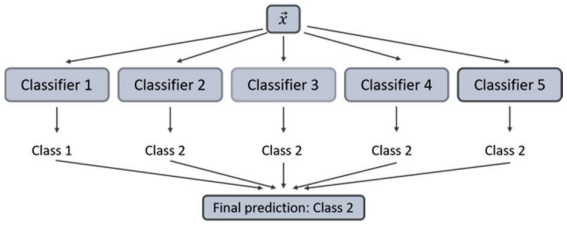

Fig. 1. Structure of Voting Classifier

3.2 BiLSTM-Deep Learning model

The major advantage of DL models over ML models is that they learn feature representations rather than the features and perform classification, in an end-to-end fashion. DL has paved the way for critical and revolutionary applications in almost every field of life in general.

DL models for text classification involve the use of WE for representing words and a learning model for the purpose of classification. WEs has been proved as a powerful representation for characterizing the statistical properties of natural language [13]. WEs are pre-trained word representations that are the key component in many neural language understanding models. However, learning high-quality representations can be challenging. They model the complex characteristics of word uses, such as syntax and semantics, and how these uses vary across linguistic contexts [14]. Since neural networks understand only numbers WE provide text to numeric vector conversion. Five popular WEs are explained below:

Continuous Bag of words (COBOW) and Skipgram model - Word2vec is a technique to produce WE for better word representation. It captures a large number of precise syntactic and semantic word relationships and represents a word in the form of a vector so that semantically similar words are grouped together and dissimilar words are located far away. There are two architectures used by Word2vec: COBOW and

Skipgram. COBOW predicts the best suited word with higher probability for a given context whereas Skipgram predicts the most appropriate context that can surround the given word. For example, given the context "Mangalore is a very [...] city", CBOW model would say that word "beautiful" or "nice" is the most probable word and words like "delightful" gets less attention. In case of Skipgram, by giving the word delightful the model would say there is a high probability that the context is "Mangalore is a very [...] city", or some other relevant context.

Global Vectors (Glove) is a global log-bilinear regression model for learning word representations that outperforms other models on word analogy, word similarity, and NER tasks. It is an extension of Word2vec method for efficiently learning word vectors [15]. GloVe constructs a word-context or word co-occurrence matrix using statistics across the entire text corpus which results in a better WE.

HPCA is a simple spectral method comparable to PCA. In the beginning, the co-occurrence matrix is normalized row-by-row to represent the words by proper discrete probability distributions. Then, the resulting matrix is transformed into a Hellinger space before applying PCA to reduce its dimensionality [3].

Fasttext is an extension of Word2vec model that represents each word as an n-gram of characters and generates embeddings by adding the character n-gram of all the n-gram representations for the words including the words that do not appear in the training corpus. The model is trained on Wikipedia texts, News corpora and on the Common Crawland is publicly available for research purpose[1].

Long Short Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) are the two popular learning models used for the purpose of classification.

LSTM -introduced by Hochreiter and Schmidhuber [16] is a special kind of Recurrent Neural Networks (RNN) capable of learning long-term dependencies. An RNN maintains a memory based on historical information which enables the model to predict the current output conditioned on long-distance features [5].

BiLSTM -networks trained using Back-Propagation-Through-Time (BPTT) [5] can efficiently use past features (via forward states) and future features (via backward states) for a specific time frame and are very effective for tagging sequential data, speech utterances or hand written documents. In LSTM, data is fed from beginning to end or in only one direction whereas in BiLSTM, data is fed in both the directions from beginning to the end and from end to beginning. BiLSTM significantly improves the accuracy of the learning model.

3.3 Transfer Learning model

TL is a ML method where the knowledge of developing a model for a specific task is reused to develop another task. It involves the concepts of a domain and a task and uses pre-trained models that have been used for one task as a base for the development of new task. TL is proved to be one of the crucial inventions in the field of DL and computer vision [17]. It stores the knowledge gained in solving the source task

[1]https://fasttext.cc/

and applies it to the development of target task [18]. Fig. 2 represents the concept of TL.

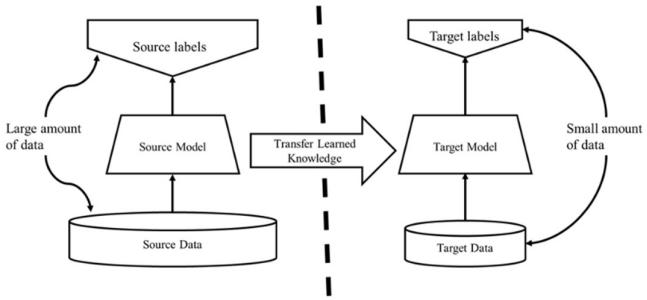

Fig. 2. Concept of Transfer Learning[2]

Low resource languages like Persian are challenging due to less/non-availability of labeled data in practice. Hence, it is difficult to train deep neural networks, as it can lead to overfit due to less training data. TL can be used as a solution in such cases where knowledge of the already trained model can be transferred to develop the new related model and the resulting model can be fine-tuned to the required task.

A Language Model (LM) is a probability distribution over sequences of words, which is used as a base model for various NLP tasks such as text classification, text summarization and text generation. Formally, "LM introduces a hypothesis space that should be useful for many other NLP tasks" [10]. For example, to build a text classification task, a learning model which is trained on a general (LM) task can be used as a base model and then text classification task can be fine-tuned. This model is able to use knowledge of the semantics of language acquired from the LM.

Universal Language Model Fine-tuning (ULMFiT) is an impressive TL method based on LM that can be applied to many NLP tasks [18]. It consists of the following stages:

i) Training the LM on a general-domain corpus that captures high-level natural language features
ii) Fine-tuning the pre-trained LM on target task data
iii) Fine-tuning the classifier on target task data

Fig. 3 describes a framework of ULMFiT model.

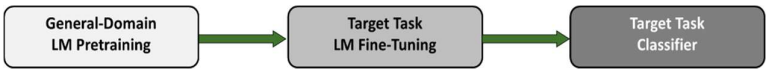

Fig. 3. A framework of ULMFiT[3]

4 Methodology

The proposed PUNER model using TL and conventional ML and DL approaches used for NER are described below:

[2] https://medium.com/the-official-integrate-ai-blog/transfer-learning-explained-7d275c1e34e2
[3]https://humboldt-wi.github.io/blog/research/information_systems_1819/group4_ulmfit/

4.1 Voting Classifier using Machine Learning approach

Three ML classifiers namely, Naïve Bayes (NB), Logistic regression (LR), and SVM are ensembled as a Voting Classifier (VC) to predict the tags/labels of NEs. The model is implemented using sklearn library using 10-fold cross validation. The test data is input to VC and based on majority voting the tag with the higher number of votes will be assigned to the input. Results of individual classifier are noted as well.

4.2 BiLSTM model using Deep Learning approach

The tagged text is given as input to the BiLSTM learning model for classification. Fig.4 illustrates a BiLSTM network for a sentence written in Persian language. The sentence "فضل به هندوستان رفت" (In English: "Fazl to India went"), is tagged as B-PER, O, B-LOC, O, where B and I tags indicate beginning and intermediate positions of NEs and O tag indicates 'Other' which means word other than a noun. (Note that in Persian, people write from right to left and also grammar is different).

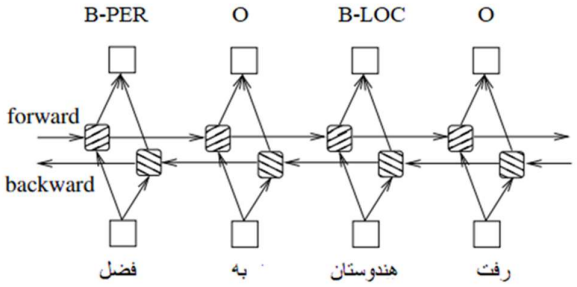

Fig.4. A bidirectional LSTM network that illustrates a Persian NER system

4.3 PUNER using Transfer Learning approach

PUNER – a Persian NER system using TL model that makes use of ULMFiT for NER in Persian language is inspired by the architecture proposed by Howard and Ruder [10]. It uses a BiLSTM model which is trained on a general LM task and then fine-tuned on NER classification task.

PUNER uses the knowledge captured from LM trained on texts collected from Wikipedia in the first stage. In this stage, LM is created using text.models module from fastai library that implements the encoder for an AWD-LSTM [19]. Once the LM completes its learning the gained knowledge is used to fine-tune the NER classification task. In the final step of model construction, obtained knowledge from LM and the training data for NER is used to train the model for tagging NEs by using an AWD-LSTM layer. Fig. 5 illustrates the architecture of PUNER.

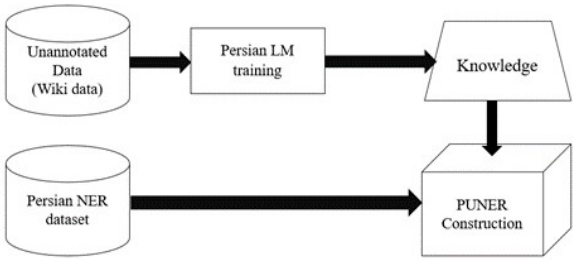

Fig. 5. Architecture of PUNER

5 Experimental Results

Evaluating the model's performance is the most important task which requires suitable dataset and suitable validation measures.

5.1 Dataset

Dataset plays a major role in the evaluation of any model's performance. In this work, a collection of unannotated Persian text is used to train the LM model for TL and two annotated Persian NER datasets namely, ArmanPersoNERCorpus [3] and Persian-NER[4] are used to validate the performance of the learning models. NE tags in the tagged datasets are in IOB format and CONLL[5] representation. IOB is the segment representation model where the tag I is assigned to intermediate NE, O to non-NE and B to the first token of consecutive NE of the same class.

Unannotated Persian text data was collected through XML dump files which include 17,000 latest Persian articles from https://dumps.wikimedia.org. Collected files have been extracted using WikiExtractor[6] module and converted to csv files such that each row contains one article. ArmanPersoNERCorpus contains six NE classes: person, organization (such as banks, teams, ministries and publishers), location (such as cities, countries, seas, and mountains), facility (such as schools, universities, hospitals, and cinemas), product (such as books, newspapers, movies, cars, theories, agreements, and religions), and event (such as wars, earthquakes, national holidays, and conferences); other is for the remaining non-NE tokens. PersianNER[7] available at https://app.text-mining.ir consists of five classes: person, organization, location, date (such as days, months, and years), and event; other is the remaining non-NE tokens. Class-wise distribution of tokens in ArmanPersoNERCorpus and PersianNER corpus is shown in Fig. 6. Statistics related to all the three datasets are given in Table 1.

[4]https://github.com/Text-Mining/Persian-NER

[5]The "CONLL" file type represents a corpus with one word per line, each word containing 10 tab-separated columns with information about the word (surface, lemma, POS, NER)

[6]https://github.com/attardi/wikiextractor

[7]PersianNER is a very big dataset. 1,000,000 tokens from the beginning part of dataset have been selected for this work.

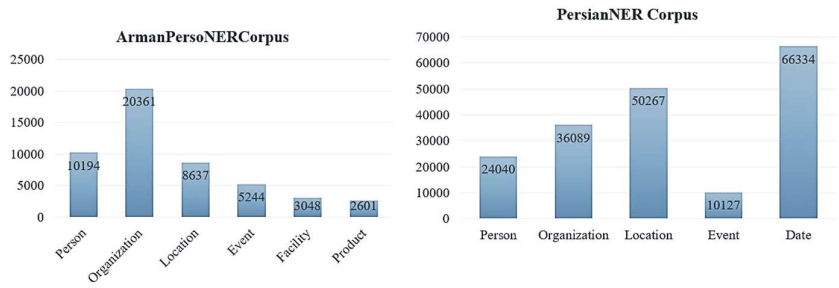

Fig.6. Annotated datasets for Persian NER

Table 1. Statistics related to dataset used for Persian NER

Dataset	**Type**	**No. Sentences**	**No. Tokens**
Wiki Texts	Unannotated	482826	18919947
ArmanPersoNER	Annotated	7,682	250,015
PersianNER	Annotated	15,363	1,000,000

5.2 Results

NER model can be dealing with important documents such as medical or legal and precise identification of NEs in those documents determines the success of the model. Therefore, the main metric to evaluate models will be F1 score (F1), Recall score (R), and Precision score (P). The results obtained in terms of F1 score, Recall score, and Precision score for all the three models are given below:

Table 2 shows the results for each ML classifier namely, NB, LR, SVM and VC which is an ensemble of NB, LR and SVM for both the tagged datasets ArmanPersoNERCorpus and PersianNER. The results illustrate that SVM classifier performs better than VC for ArmanPersoNERCorpus dataset whereas VC gives better results for PersianNER. DL approach is implemented using TensorFlow-Keras BiLSTM model with five different WEs namely, Fasttext, Skipgram, HPCA, Glove and COBOW. Each session is run for 10 epochs. Table 3 shows the results for BiLSTM model for five different WEs for both the tagged datasets ArmanPersoNERCorpus and PersianNER. For ArmanPersoNERCorpus, the results of BiLSTM model range from a minimum of 88.75 F1 score for HPCA WE to a maximum of 93.60 F1 score for COBOW WE. For PersianNER dataset, the results of BiLSTM model range from a minimum of 69.25 F1 score for Skipgram WE to a maximum of 71.55 F1 score for Fasttext WE. Table 4 illustrates the results of the proposed model PUNER on both datasets ArmanPersoNERCorpus and PersianNER. PUNER gives 92.82 F1 score for ArmanPersoNERCorpus and 82.16 F1 score for PersianNER corpus.

Table 2. Results of ML model

Methods	ArmanPersoNER			Persian-NER		
	Precision	Recall	F1	Precision	Recall	F1
Naïve Bayes	29.64	49.28	31.53	44.72	70.29	51.70
Logistic Regression	72.34	39.68	51.25	74.85	45.78	56.81
SVM	70.11	51.78	**57.67**	27.75	68.73	34.21
Ensemble (NB & LR & SVM)	73.01	44.53	55.32	74.78	49.49	**59.56**

Table 3. Results of BiLSTM model using different word embeddings

Word Embeddings	ArmanPersoNER			Persian-NER		
	Precision	Recall	F1	Precision	Recall	F1
Fasttext	86.47	91.80	89.06	70.85	72.26	**71.55**
Skipgram	91.47	93.05	92.26	67.09	71.56	69.25
HPCA	87.21	90.34	88.75	67.27	71.86	69.49
Glove	91.07	92.06	91.56	69.69	72.75	71.19
COBOW	93.39	93.82	**93.60**	71.41	71.45	71.43

Table 4. Results of PUNER

Methods	ArmanPersoNER			Persian-NER		
	Precision	Recall	F1	Precision	Recall	F1
PUNER	92.72	93.44	92.82	82.02	84.42	82.16

A comparison of all the three learning models namely ML, DL and PUNER in terms of Precision, Recall, and F1 score is shown is Table 5. The F1 score in results illustrate that PUNER performs better on PersianNER dataset and BiLSTM using COBOW WE gives better result for ArmanPersoNERCorpus. Further, F1 score of PUNER for ArmanPersoNERCorpus is close to BiLSTM using COBOW WE. DL approach has illustrated higher results than that of ML and TL for ArmanPersoNERCorpus dataset. This relative result seems to be good according to other NER results in Persian and other languages from the literature. However, on the average PUNER performs better than conventional ML model and also the DL model. A graphical representation of the comparison of F1 scores of all the three models on the two datasets is shown in Fig. 7.

6 Conclusion

This paper presents PUNER - a Persian NER system using Transfer Learning (TL) model that makes use of ULMFiT for NER in Persian language. PUNER is initially trained on a general domain data collected from Wikipedia, and then it is applied on target NER data. The model is evaluated on two annotated datasets namely ArmanPersoNERCorpus and PersionNER. Results show that the proposed model has achieved 92.82 F1 score on ArmanPersoNERCorpus 82.16 F1 score on Persian-NER dataset. For the sake of comparison, Voting Classifier which is an ensemble of ML

approach using Naïve Bayes, Logistic Regression, and SVM algorithms and BiLSTM model using five different WEs namely, Fasttext, Skipgram, HPCA, Glove and COBOW has been implemented. The comparison of all the approaches illustrate that PUNER definitely performs better than ML models and on par with DL models. On the average, PUNER gives the result which is on par with DL models. Further, the ULMFiT Persian LM weights built for Persian NER can be utilized for other Persian NLP tasks.

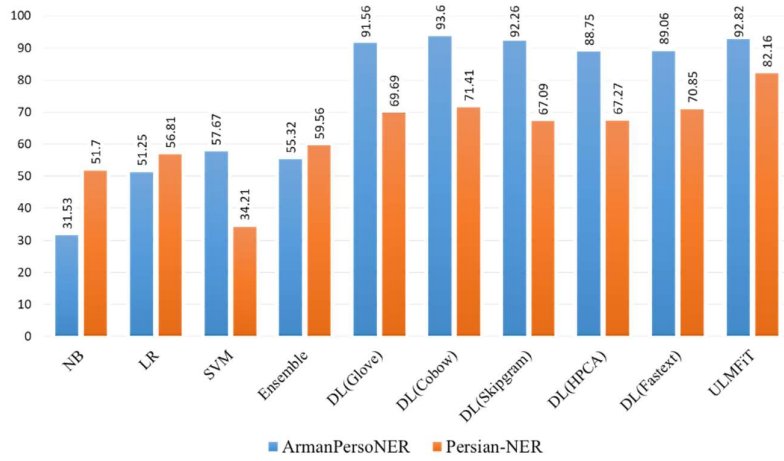

Fig. 7. F1 score comparison of all models

Table 5. Comparison of learning models for Persian NER

Approach	Methods	ArmanPersoNER			Persian-NER		
		Precision	Recall	F1	Precision	Recall	F1
ML	Naïve Bayes	29.64	49.28	31.53	44.72	70.29	51.70
	Logistic Regression	72.34	39.68	51.25	74.85	45.78	56.81
	SVM	70.11	51.78	57.67	27.75	68.73	34.21
	Ensemble (NB & LR & SVM)	73.01	44.53	55.32	74.78	49.49	59.56
DL	BiLSTM (Fasttext)	86.47	91.80	89.06	84.15	84.16	70.85
	BiLSTM (Skipgram)	91.47	93.05	92.26	70.85	72.26	67.09
	BiLSTM (HPCA)	87.21	90.34	88.75	67.09	71.56	67.27
	BiLSTM (Glove)	91.07	92.06	91.56	67.27	71.86	69.69
	BiLSTM (Cobow)	93.39	93.82	**93.60**	69.69	72.75	71.41
TL	ULMFiT	92.72	93.44	92.82	82.02	84.42	**82.16**

References

1. Amarappa, S., and S. V. Sathyanarayana. 'Kannada Named Entity Recognition and Classification (NERC) Based on Multinomial Naive Bayes (MNB) Classifier'. ArXiv preprint arXiv: 1509.04385, 2015.

2. Poostchi, Hanieh, Ehsan Zare Borzeshi, Mohammad Abdous, and Massimo Piccardi. ‘Personer: Persian Named-Entity Recognition’. In COLING 2016-26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers, pp. 3381-3389, 2016.
3. Poostchi, Hanieh, Ehsan Zare Borzeshi, and Massimo Piccardi. ‘BiLSTM-CRF for Persian Named-Entity Recognition ArmanPersoNERCorpus: The First Entity-Annotated Persian Dataset’. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 4427-4431, 2018.
4. Khormuji, Morteza Kolali, and Mehrnoosh Bazrafkan. ‘Persian Named Entity Recognition Based with Local Filters’. International Journal of Computer Applications 100, no. 4, 2014.
5. Huang, Zhiheng, Wei Xu, and Kai Yu. ‘Bidirectional LSTM-CRF Models for Sequence Tagging’. ArXiv preprint arXiv: 1508.01991, 2015.
6. Indra, S. T., Liza Wikarsa, and Rinaldo Turang. ‘Using Logistic Regression Method to Classify Tweets into the Selected Topics’. In 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, pp. 385-390. 2016.
7. Taghizadeh, Nasrin, Zeinab Borhanifard, Melika Golestani Pour, and Heshaam Faili. ‘NSURL-2019 task 7: Named Entity Recognition (NER) In Farsi’. ArXiv preprint arXiv: 2003.09029, 2020.
8. Mahdi Mohseni and Amirhossein Tebbifakhr. ‘Morphobert: A Persian NER System with BERT and Morphological Analysis’. In Proceedings of the First International Workshop on NLP Solutions for Under Resourced Languages, NSURL ’19, Trento, Italy, 2019.
9. Mahalakshmi, G. S., J. Antony, and S. Roshini. ‘Domain Based Named Entity Recognition Using Naive Bayes Classification’. Aust. J. Basic & Appl. Sci., 10(2): pp.234-239, 2016.
10. Howard Jeremy, and Sebastian Ruder. ‘Universal Language Model Fine-Tuning for Text Classification’. ArXiv preprint arXiv: 1801.06146, 2018.
11. Nayel, Hamada A., H. L. Shashirekha, Hiroyuki Shindo, and Yuji Matsumoto. ‘Improving Multi-Word Entity Recognition for Biomedical Texts’. ArXiv preprint arXiv: 1908.05691, 2019.
12. Pham, Thai-Hoang, Khai Mai, Nguyen Minh Trung, Nguyen Tuan Duc, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. "Multi-Task Learning with Contextualized Word Representations for Extended Named Entity Recognition." arXiv preprint arXiv: 1902.10118, 2019.
13. Wang, Peilu, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. ‘Part-Of-Speech Tagging With Bidirectional Long Short-Term Memory Recurrent Neural Network’. ArXiv preprint arXiv: 1510.06168, 2015.
14. Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. ‘Deep Contextualized Word Representations’. ArXiv preprint arXiv: 1802.05365, 2018.
15. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 'Glove: Global Vectors for Word Representation', In Proceeding DXs of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543., 2014.
16. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8, pp. 1735-1780, 1997.
17. Saha, Rohan, ‘Transfer Learning - A Comparative Analysis’, 2018.
18. Faltl, Sandra, Michael Schimpke, and ConstantinHackober. ‘ULMFiT: State-of-The-Art in Text Analysis’, 2019.
19. Merity, Stephen, Nitish Shirish Keskar, and Richard Socher. ‘Regularizing and Optimizing LSTM Language Models’. ArXiv preprint arXiv: 1708.02182., 2017.