BERT-PersNER: a New Model for Persian Named Entity Recognition

Farane Jalali Farahani
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
fjalali@ce.sharif.edu

Gholamreza Ghassem-Sani
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
sani@sharif.edu

Abstract

Named entity recognition (NER) is one of the major tasks in natural language processing. A named entity is often a word or expression that bears a valuable piece of information, which can be effectively employed by some major NLP tasks such as machine translation, question answering, and text summarization. In this paper, we introduce a new model called BERT-PersNER (BERT based Persian Named Entity Recognizer), in which we have applied transfer learning and active learning approaches to NER in Persian, which is regarded as a low-resource language. Like many others, we have used Conditional Random Field for tag decoding in our proposed architecture. BERT-PersNER has outperformed two available studies in Persian NER, in most cases of our experiments using the supervised learning approach on two Persian datasets called Arman and Peyma. Besides, as the very first effort to try active learning in the Persian NER, using only 30% of Arman and 20% of Peyma, we respectively achieved 92.15%, and 92.41% performance of the mentioned supervised learning experiments.

1 Introduction

Named Entity Recognition (NER) is a fundamental Natural Language Processing (NLP) task, in which we try to identify and classify certain entities such as organizations, persons, locations, etc. in a given text. A named entity is often a word or expression in the text that bears a valuable piece of information, which can be effectively used in other high-level NLP tasks such as machine translation, question answering, and text summarization. Unlike English, which is rich in digital resources, there are some natural languages such as Persian, which is regarded to be a low-resource language. However, Persian is an important language because it is spoken by more than 110 million people worldwide.

In this article, we introduce an efficient model for Persian NER. To tackle problems such as the lack of labeled data in Persian, we have used two different learning methods, i.e., transfer learning (Pan and Yang, 2010) and active learning (Settles, 2010). To transfer knowledge in transfer learning, the model-based (parameter) approach (Settles, 2010) has been used to fine-tune BERT (Devlin et al., 2019). Through transfer learning, we can produce a powerful model using limited data that takes much less training time. We also employed active learning in order to perform the fine-tuning task by using only a few informative samples instead of the whole dataset. Through active learning, it is possible to deliver a performance that is very close to that of a supervised learning method.

The structure of this paper is as follows. We review the literature in Sec. 2; BERT is discussed in Sec. 3; Our proposed method is explained in Sec. 4; The results of our experiments are given in Sec. 5 and the article is concluded in Sec. 6.

2 Literature Review

2.1 Named Entity Recognition

The models that have tackled NER through Deep Learning (DL) consist of three main components: 1) input representation, 2) context encoder, and 3) tag decoder (Li et al., 2020). For the first component, Word2Vec (Mikolov et al., 2013) was used in (Lample et al., 2016) and both Word2Vec and GloVe (Pennington et al., 2014) were used in (Poost-

chi et al., 2018). As the second component, in most studies, a bidirectional long short-term memory (BiLSTM) has been used to capture long-distance dependencies (Li et al., 2020; Shahshahani et al., 2019; Lample et al., 2016; Bokaei and Mahmoudi, 2018; Poostchi et al., 2018). In other studies, an architecture named Transformer (Vaswani et al., 2017) was used. In the architecture of Transformer, there is no recurrent structure and it operates based on attention mechanisms, which lead to an increase in parallelization (Vaswani et al., 2017). The Pre-trained BERT model is based on the Transformer architecture and supports more than 100 live languages, including Persian. It has also been applied to NER (Devlin et al., 2019; Taher et al., 2019). As the last component, Conditional Random Field (CRF) (Lafferty et al., 2001) has been mostly used as the referred tag decoder. The main reason for choosing CRF is that, instead of merely looking for the best label (tag) for each word, it jointly uses neighboring tags to determine a sequence of output labels (Li et al., 2020; Lample et al., 2016; Bokaei and Mahmoudi, 2018; Poostchi et al., 2018; Taher et al., 2019).

Dataset	Entity type	#Tokens	%
Arman	Person	5215	2.08
	Organization	10036	4.01
	Location	4308	1.72
	Facility	1485	0.59
	Product	1463	0.58
	Event	2518	1
	Other	224990	89.99
Peyma	Person	7675	2.53
	Organization	16964	5.6
	Location	8782	2.90
	Time	732	0.24
	Date	4259	1.4
	Money	2037	0.67
	Percent	699	0.23
	Other	261382	86.39

Table 1: Details of Arman and Peyma datasets (Poostchi et al., 2018; Shahshahani et al., 2019).

2.2 Dataset

One of the datasets that have been used in Persian NER is called Arman, which was first published in 2016. It consists of about 250k tokens and six classes: location, organization, person, facility, product, and event (Poostchi et al., 2018) [1]. In 2018, another dataset called Peyma was published in (Shahshahani et al., 2019), which includes 300k tokens and seven classes: location, organization, person, time, data, money, and percent, Table 1 [2].

2.3 Active Learning

The active learning strategies are divided into three groups: pool-based sampling, stream-based sampling, and membership query synthesis (Settles, 2010). In the first two, the instances are taken from a pool of data and a stream of data, respectively. However, in the last one, instances are generated. The main advantage of the pool-based sampling is that it provides the possibility of running a comparison among all instances and selecting the most informative samples for training the model. The pool-based sampling method has been applied to English NER in several studies (Shen et al., 2017; Chen et al., 2015).

In (Shen et al., 2017), OntoNotes-5.0 English and Chinese were used for active learning experiments. The authors employed different selection strategies, where in each case, an LSTM based model was firstly trained with 1% of the original training dataset. Then, in each round, the most informative instances were selected from the remaining 99% for training; and each round was ended when 20,000 words had been added to the training dataset. The training process was repeated at the end of each round based on the accumulated dataset and the parameters of the model were updated through this repetition of training. They showed that by this active learning process, one could achieve 99% performance of supervised learning, using only 30.1% of the Chinese dataset. It was also shown that the same performance could be achieved using only 24.9% of the English dataset.

In (Chen et al., 2015), the NER task was handled for medical texts. The authors used

[1] Arman is available at: https://github.com/HaniehP/PersianNER/blob/master/ArmanPersoNERCorpus.zip

[2] Peyma is available via a folder named 300K at: http://en.itrc.ac.ir/sites/default/files/pictures/NER.rar

the 2010 i2b2/VA annotated dataset. They randomly split the dataset into two parts: (1) a pool including 80% of the data for being used during the active learning process and (2) a test set including the remaining 20% for assessing the NER model. The authors simulated practical pool-based active learning by getting labels from the mentioned pool instead of interacting with an actual user. It was emphasized that the labels had not been accessed unless the active learning algorithm selected an instance for being added to the training data. They gained F1 score of 80% on the 2010 i2b2/VA dataset, using only 58% of the original training data. To the best of our knowledge, active learning has not been previously applied to Persian NER and our work is the first examination of this learning method in Persian NER.

3 BERT

As it was mentioned before, the architecture of BERT is based on Transformer. BERT is a pre-trained model, trained on about 3.3 billion unlabeled data. Its input embedding consists of three parts: 1) token embedding, 2) segment embedding, and 3) position embedding (Devlin et al., 2019). For token embedding, WordPiece has been employed. Segment embedding is used to distinguish a pair of input sentences. The goal of position embedding is to determine sentence word order as the words of each sentence are fed to BERT simultaneously and without any specific order (Devlin et al., 2019).

BERT is applicable in both freezing and fine-tuning methods (Pan and Yang, 2010). In freezing, no change is applied to the pre-trained model. Freezing leads to the extraction of constant features from the model so that they can be used as contextualized word embeddings. In contrast, in the fine-tuning method, the parameters of certain layers are fine-tuned based on a down-stream task and labeled data. It was shown that fine-tuning outperforms freezing (Peters et al., 2019). Therefore, we have used fine-tuning in our proposed method. Besides, like several previous studies, we have used CRF for the tag decoding. The detailed account of our proposed method, which we are going to refer to as BERT-PersNER, is given next.

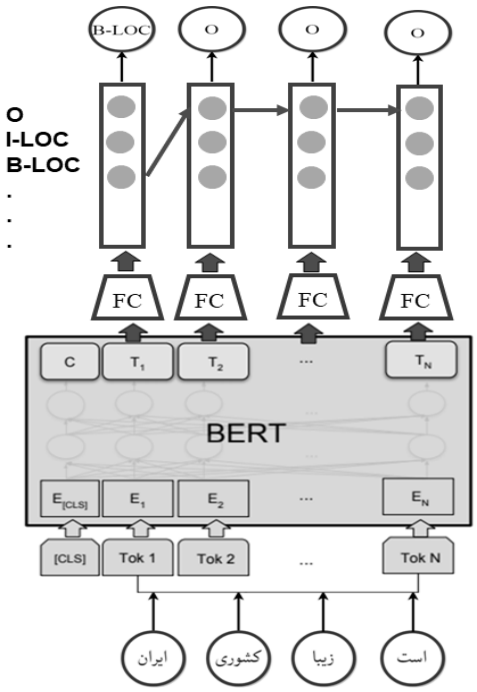

Figure 1: The figure of our proposed model with an example. The input sequence is a Persian sentence meaning *Iran is a beautiful country*. Token *Iran* is the first token in the input sentence and is predicted as *B-LOC*.

4 Proposed Method

4.1 BERT-PersNER

BERT has been used as the first two sections of our NER architecture (i.e., the input representation and the context encoder sections). BERT generates an intermediate representation for each token. These intermediate representations will be later used to predict the output label sequences. As the tag decoder section of the mentioned NER architecture, we have used CRF for predicting output labels. We have used a number of fully connected neural networks to ensure that the BERT output vector dimension will match with the number of possible tags for each token. The architecture of BERT-PersNER is shown in Figure 1.

Assuming that $x = \{x_1, x_2, \cdots, x_N\}$ shows the observed input tokens of length N, and $y = \{y_1, y_2, \cdots, y_N\}$ represents the corresponding output labels in a linear-chain CRF, P(y|x) is calculated as follows:

$$P(y|x) = \frac{e^{Score(x,y)}}{\sum_{y' \in Y(x)} e^{Score(x,y')}}, \qquad (1)$$

where

$$Score(x, y) = \sum_{i=0}^{N} T_{y_i, y_{i+1}} + \sum_{i=1}^{N} P_{i, y_i} \quad (2)$$

In equation 2, the transition matrix $T \in R^{K+2 \times K+2}$, in which K is the count of distinguished labels and +2 is added for the beginning and ending labels, represents a transformation from one label to another. The transition matrix is initialized randomly and is updated during training. The fully connected neural network output is in the P matrix, where P_{i,y_i} represents the score of label y_i for i^{th} input token. Y(x) is the set of all possible labeled sequences for x. The loss function equals the negative log-likelihood and the best sequence label obtained through the following equation and the Viterbi algorithm:

$$y^* = \underset{y' \in Y(x)}{\operatorname{argmax}} \, log P(y'|x) \quad (3)$$

4.2 Active Learning in BERT-PersNER

In this work, similar to some earlier researches (Shen et al., 2017; Chen et al., 2015), we have simulated the pool-based sampling method of active learning and, instead of interacting with an actual user for obtaining labels, an annotated dataset has been used. Akin to some earlier works, we randomly divided the original training dataset into two parts: an initial batch named L, which contained 1% of the data, and a pool named U, which included the remaining 99%. Note that the label of no instance in U was accessed unless that our selection strategy chose that instance for being added to L.

The active learning framework in BERT-PersNER consists of the following four steps:

1. Building the initial model: to begin with, the initial BERT-PersNER model is obtained through performing the training process using the L batch.

2. Sorting the instances: at this stage, based on a selection strategy, the informativeness of the instances in U is measured (i.e., done without accessing the labels). Then, 10% of the top-ranked instances, together with their labels, are removed from U and added to L.

3. Training: the BERT-PersNER model is then trained again on the new L and the parameters of the model are updated.

4. Iterating: stages 2 and 3 are repeated until that the instances of U are exhausted.

The selection strategy can be regarded as the most important part of active learning, because the informativeness of each instance is determined by this strategy. In this study, uncertainty sampling (Lewis and Gale, 1994) has been used as our selection strategy. This selection strategy is based on the idea that if the label of those instances on which the model is less certain is known to the system, it will be more beneficial. The following three uncertainty sampling strategies have been implemented in this study:

- Normalized Least Confidence (NLC) (Lewis and Gale, 1994): in this strategy, the certainty of the best label sequence for each input sample is used as a criterion to find the least confident instances. To eliminate the length effect of input sentences, for each instance, a Normalized form of least confidence is calculated through the following equation:

$$\phi^{NLC}(x) = 1 - \frac{1}{N} P(y^*|x; \theta), \quad (4)$$

  in which x is the input sequence, y^* represents the best label sequence for x, N is the length of x, θ shows the model parameters, and P is the instance confidence (i.e., based on equation 1).

- Margin (M) (Scheffer et al., 2001): in this strategy, using the following equation, the marginal difference between the first two best label sequences is used as the criterion for finding the least confident instances:

$$\phi^{M}(x) = -(P(y_1^*|x; \theta) - P(y_2^*|x; \theta)), \quad (5)$$

  in which the negative sign is added to select the instances with the lowest margin.

- Sequence Entropy (SE) (Settles and Craven, 2008): here, the entropy of all labels sequences is used as the criterion to

find the least confident instances through the following equation:

$$\phi^{SE}(x) = -\sum_{y'} P(y'|x;\theta) log P(y'|x;\theta) \quad (6)$$

Since the number of possible label sequences grows exponentially as our input sentence length increases, we have only considered the N-best (with N = 16) label sequences for each instance.

5 Experiments

Dataset	Entities	Word-level		Phrase-level
		B-	I-	
Arman	Person	92.26	93.59	90.52
	Organization	81.61	87.97	79.43
	Location	82.08	78.67	81.83
	Facility	75.62	80.78	69.82
	Product	70.95	75.30	65.89
	Event	70.95	78.83	60.44
	All classes	84.23		80.80
Peyma	Location	86.78	76.02	84.89
	Person	86.88	91.19	84.10
	Organization	83.09	87.23	78.29
	Time	75.90	82.72	70.96
	Date	84.23	86.91	81.38
	Money	92.52	92.01	81.72
	Percent	91.64	94.48	89.32
	All classes	86.14		82.05

Table 2: F1 scores (in percentage) of running our model on Arman and Peyma.

Work	Arman		Peyma	
	Word-level	Phrase-level	Word-level	Phrase-level
Deep-CRF (Bokaei and Mahmoudi, 2018)	81.50	76.79	N/A	N/A
(Shahshahani et al., 2019)	N/A	N/A	**87**	80
BERT-PersNER	**84.23**	**80.80**	86.14	**82.05**

Table 3: A comparison between F1 scores (in percentage) of BERT-PersNER and our baselines.

In this section, the results of applying supervised learning and active learning methods to two Persian NER datasets, Arman and Peyma, are presented. These datasets have been tagged based on the IOB format. In all our experiments, BERT-base-multilingual-cased has been employed with a learning rate of $5e^{-5}$ and a maximum sequence length of 180. Besides, the learning rate for CRF has been set to $8e^{-5}$, and the batch size has been set to 8. To run the experiments on Arman, the same dividing configuration as the one published on GitHub has been used. That is, Arman is divided into 3 equal portions. We have also divided Peyma into 3 equal parts. For each dataset, the 3-fold cross-validation has been implemented and the results have been averaged. In the case of supervised learning, the evaluation has been performed on both word- and phrase-level. The result of BERT-PersNER in supervised learning experiments is shown in Table 2.

Word-level evaluation of BERT-PersNER on Arman shows that the best performance of this model is achieved on *I-person*, and its weakest performance is where it deals with *B-event* and *B-product*. On the other hand, in the phase-level evaluation, the model performs best on *Person* and worst on *Event*. This is mainly due to the differences between tag counts in Arman; as it is depicted in Table 1, the frequency of *Person* tags is twice as many as *Event* tags, and four times as that of *Product*. Note that although *Location* is the most frequent tag in Arman, its resemblance to *Organization* causes that the BERT-PersNER performance on *Location* to become weaker than its performance on *Person*.

Word-level evaluation of BERT-PersNER on Peyma shows that its best performance is on *I-percent* and its worst performance is on *B-time*. On the other hand, phrase-level evaluation of the model indicates that it works best on *Percent* and its weakest performance is on *Time*. This is due to the fact that, although the tag count of *Percent* in Peyma is not high, the low level of variety between different *Percent* tags allows the model to learn this class effectively. On the other hand, due to a low tag count of *Time* and resemblance to *Date* make BERT-PersNER acts worst on *Time*.

Dataset	Selection Strategy	Percent of training data								
		10	20	30	40	50	60	70	80	90
Arman	NLC	70.40	75.75	77.62	79.19	79.68	80.67	81.34	81.52	81.64
	M	68.91	73.61	76.17	76.89	77.85	79.16	79.21	79.90	80.45
	SE	71.84	75.85	77.07	77.98	78.74	79.27	79.67	80.28	81.23
	RAND	65.89	72.82	75.84	77.11	77.44	78.18	79.20	80.95	81.30
Peyma	NLC	75.60	79.61	81.16	82.00	82.69	83.09	83.03	83.21	83.22
	M	72.68	76.52	79.65	80.56	81.59	82.08	82.62	82.83	82.88
	SE	72.46	78.14	80.05	80.79	81.53	82.03	82.34	82.92	83.15
	RAND	72.26	77.35	79.13	80.46	81.09	81.44	81.95	82.40	82.78

Table 4: A comparison between F1 scores (in percentage) of different selection strategies in active learning.

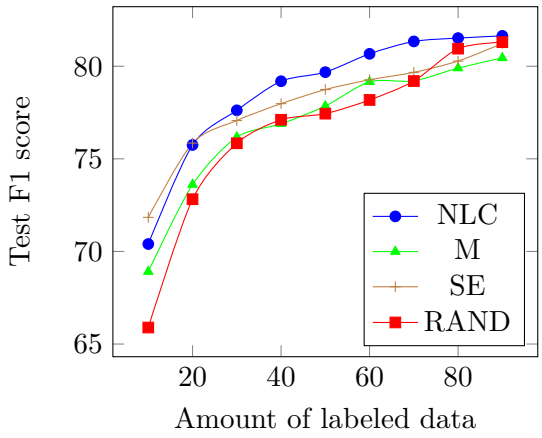

Figure 2: BERT-PersNER performance on Arman, using different selection strategies

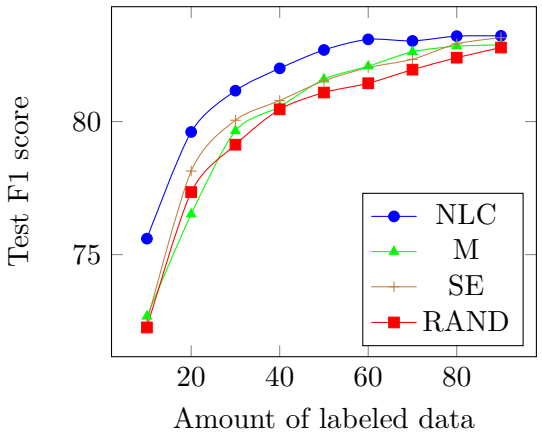

Figure 3: BERT-PersNER performance on Peyma, using different selection strategies

Table 3 shows BERT-PersNER performance against two recent studies (Shahshahani et al., 2019; Bokaei and Mahmoudi, 2018). As it is observed, our proposed model outperforms the baselines on Arman. More precisely, word-level evaluation, BERT-PersNER improved Deep-CRF by 2.73%. Furthermore, in phrase-level evaluation, it enhanced the results of the baseline by 4.01%.

As it is shown in 3, a word-level evaluation of BERT-PersNER performance on Peyma against that of (Shahshahani et al., 2019) did not result in any improvement. However, in a phrase-level evaluation, it has improved the result of baseline by 2.05% of F1 score.

In all our active learning experiments, the evaluation of BERT-PersNER has been performed in word-level. Since a gradual increase of data on its own can improve the model performance, in addition to the previously mentioned selection strategies, we have used a random selection strategy as our baseline. Here, too, a three-fold cross validation has been employed and the final results have been averaged over the three-fold. In each iteration, the chosen selection strategy takes 10% of the remaining unlabeled data from the pool and adds it to the training dataset. That is about 507 and 546 sentences of Arman and Peyma, respectively.

As it can be observed in Figures 2 and 3, reflecting the results of Table 4, NLC has outperformed its counterparts. We think this is due to the fact that it elects the instances based on their very best label sequence, whereas other strategies also take other slightly weaker instances into consideration (e.g., the second best label sequence is used in Margin). As it is shown in Figure 2, on average, the performance of SE, M, and RAND on Arman has been respectively weaker than that of RAND. On Peyma (cf. Figure 3), again, RAND has been the weakest of all; but there has not been any clear superior between SE or M.

As it can also be seen in the mentioned Figures, in the earlier stages of active learning, the gradient is much higher than its later stages, which is an indication of selecting much more informative instances in those earlier stages.

Using only 30% of Arman, NLC achieved 92.15% performance of supervised learning. On the other hand, in the case of Peyma, using 20% of data, NLC reached 92.41% performance of the supervised learning approach. Therefore, by using more informative instances, we can reach a performance compatible with that of supervised learning with much less required data. This data saving is particularly critical in the case of low-recourse languages such as Persian.

6 Conclusion

We fine-tuned the pre-trained BERT model for the task of Named Entity Recognition in Persian, which is regarded as a low-resource language. We employed both transfer learning and active learning methods to develop a new model called BERT-PersNER for Persian NER. The new model was evaluated on two Persian NER datasets, which are called Arman and Peyma. We first evaluated BERT-PersNER in a supervised approach, which was done on both word- and phrase-level. Our new model outperformed two previous studies in Persian NER by 2.05%, 2.73%, and 4.01% F1 scores, in phrase-level on Peyma, in word-level on Arman, and in phrase-level on Arman, respectively. In the word-level evaluation on Peyma, however, the performance of BERT-PersNER was lower than the best available related work by 0.86% F1 score.

BERT-PersNER was evaluated word-level using the active learning approach with four different selection strategies. As our main contribution, it was shown that using only 30% of Arman, we can achieve 92.15% performance of the supervised learning method. It was also shown that using only 20% of Peyma, we can reach 92.41% performance of the supervised learning case. To the best of our knowledge, the application of active learning in Persian NER is the very first effort in this respect. As our future work, we intend to investigate the impact of other selection strategies on BERT-PersNER. We also plan to evaluate the proposed approach using other newly published pre-trained models.

References

Mohammad Hadi Bokaei and Maryam Mahmoudi. 2018. Improved deep persian named entity recognition. In *2018 9th International Symposium on Telecommunications (IST)*.

Yukun Chen, Thomas Lasko, Qiaozhu Mei, Joshua Denny, and Hua Xu. 2015. A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc.

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*. Association for Computational Linguistics.

David Lewis and William Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Educational Activities Department*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors

for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Matthew Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Association for Computational Linguistics.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. 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)*. European Language Resources Association (ELRA).

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*. Springer-Verlag.

Burr Settles. 2010. Active learning literature survey. Technical report, University of Wisconsin, Madison.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam and Faili. 2019. Payma: A tagged corpus of persian named entities. *Signal and Data Processing*.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Association for Computational Linguistics.

Ehsan Taher, Seyed Abbas Hoseini, and Mehrnoush Shamsfard. 2019. Beheshti-NER: Persian named entity recognition using BERT. In *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019 - Short Papers*. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.