Journal of Artificial Intelligence and Data Mining (JAIDM), x(x): xxx-xxx, xxxx

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

ParsNER-Social: A Corpus for Named Entity Recognition in Persian Social Media Texts

Majid Asgari-Bidhendi, Behrooz Janfada, Omid Reza Roshani Talab, Behrouz Minaei-Bidgoli[1*]

Computer Engineering School, Iran University of Science and Technology, Tehran, Iran

Article Info

Article History:
Received 10 September 2020
Revised 12 October 2020
Accepted 25 November 2020

DOI:10.22044/jadm.2020.9949.2143

Keywords:
Named Entity Recognition, Natural Language Processing, Social Media Corpus, Persian Language.

**Corresponding author: b_minaei@iust.ac.ir(B. Minaei-Bidgoli).*

Abstract

Named Entity Recognition (NER) is one of the essential prerequisites for many natural language processing tasks. All public corpora for Persian named entity recognition such as ParsNERCorp and ArmanPersoNERCorpus are based on the Bijankhan corpus, which is originated from the Hamshahri newspaper in 2004. Correspondingly, most of the published named entity recognition models in Persian are specially tuned for the news data and are not flexible enough to be applied in different text categories such as social media texts. In this work, we introduce ParsNER-Social, a corpus for training named entity recognition models in the Persian language built from social media sources. This corpus consists of 205,373 tokens, and their NER tags crawled from social media contents, including 10 Telegram channels in 10 different categories. Furthermore, three supervised methods are introduced and trained based on the ParsNER-Social corpus: two conditional random field models as baseline models and one state-of-the-art deep learning model with six different configurations are evaluated on the basis of the proposed dataset. The experiments performed show that the Mono-Lingual Persian models based on Bidirectional Encoder Representations from Transformers (MLBERT) outperform the other approaches on the ParsNER-Social corpus. Among the different configurations of the MLBERT models, the ParsBERT+BERT-TokenClass model has obtained an F1-score of 89.65%.

1. Introduction

Named Entity Recognition (NER) is a crucial task in natural language processing (NLP). NER is focused on automatically extracting the proper names from the text and classifying them to pre-defined categories such as person names, location names (such as cities, countries, and rivers), and organization names (such as companies and organizations). NER is one of the essential components for many natural language tasks, including relation extraction, text summarization [1], information extraction, machine translation, question answering, and semantic search. The term named entity was introduced in the sixth message understanding conference in 1995 [2].

There are high-accuracy results for NER in the resource-rich languages such as English and French since the last two decades [3], whereas there are no convincing results for the resource-poor languages such as Persian yet. There were very few works on NER in the Persian language due to the lack of useful publicly-available corpora before 2015. ParsNERCorp[1] was the first notable NER corpus that was created in 2012 and published with an open-source named entity recognizer in 2016. In the same year, Poostchi et

[1] The code and the corpus are available at https://github.com/majidasgari/ParsNER

al. published a 6-class NER corpus named ArmanPersoNERCorpus [4].

Currently, texts on social media are among the most expensive natural language textual sources in the world. We live amid a massive amount of social media data.

Much of the information in the social media texts are in the form of natural language and are full of named entities. The analysis of this abundant and continuous flow of user-generated content could provide valuable information that was not possible through traditional media texts. Since data-annotating is time-consuming and costly, there are few data sets for NLP tasks for resource-poor languages such as Persian. Most NER research works focus on structured datasets such as CoNLL [5]. However, processing unstructured texts (such as social media texts) is a more complicated task and is required in the real world. As a public domain, not a domain-specific task, NER must identify named entities in both the structured data and social media contents. A general domain NER should also be able to identify new named entities with new forms. However, the lack of unstructured datasets with standard gold annotations is a pinch in creating the NER methods for unstructured texts. Moreover, such datasets have never been developed for Persian as a low-resource language [2,6,7].

In this paper, we introduce ParsNER-Social, a hand-annotated corpus created from the social media data crawled from 10 Telegram channels. Three state-of-the-art supervised methods have been trained on this corpus, and the results obtained are published in this paper. The rest of this paper is structured as follows. Section 2describes the significant efforts made for NER in the English and Persian languages and also NER on the social media texts; well-known datasets in English are also introduced in this section. Section 3 introduces the ParsNER-Social corpus. In Section 4, we present the experimental results that have been obtained from the state-of-the-art methods on the corpus. Finally, in the last two sections, we specify the released resources, conclude this paper, and propose future works.

2. Related Works

In this section, we briefly review the works related to this research work. We first introduce the well-known corpora for the rich-resource languages. Then we will deal with the most significant works in NER in English and will review the related works in Persian.

2.1. Well-known NER Datasets

Three main corpora have been used to train and evaluate the NER methods:

- CoNLL-2003 shared task corpus [5], which was the first notable corpus for NER for the English and German languages. This corpus consists of newswire articles from the Reuters RCV1 corpus. CoNLL-2003 is annotated for four classes, including Person, Location, Organization, and Miscellaneous.
- OntoNotes is a sizeable multilingual training corpus, including parse trees, relation extraction, word sense disambiguation, and name types [8]. OntoNotes corpus includes 18 NER tags, consisting of eleven types (such as organization and location) and seven values (such as percent and date).
- WNUT2017 shared a task that focuses on the unusual and unseen entities, primarily in the context of emerging discussions [9]. WNUT2017 is annotated for six classes, including location, person, group, corporation, product, and creative work.

Of these three corpora, various studies have used the oldest NER corpus, CoNLL2003, the most for their models' training and evaluations.

2.2. NER in English Language

In recent years, studies on NER have been dominated by deep learning approaches [10]. Also, in specific-domain NER, the methods utilizing deep neural networks still play a leading role [11,12]. Moreover, in the studies related to NER in low-resource languages, the deep neural network-based methods have attracted the most attention [6,13].In a new study, Yadav and Bethard have reviewed many NER models and architectures [14], showing that in recent years, the methods based on deep neural networks in the field of NER are superior to the other models. They have categorized the NER systems into three classes: knowledge-based systems (such as [15]), unsupervised and bootstrapped systems (such as [16]), and feature-engineered supervised systems (such as [17,18]); and then compared their results with the feature-inferring neural network systems [19].

Xin et al. [20] have proposed IntNet, a convolutional neural architecture that learns representations of the internal structure of words by composing their characters from a limited corpus. Most models based on embedding vectors in NER use word or sentence embeddings. Although considerable research works have been done in the field of learning character embedding, it is not yet clear, which is the best type of

architecture for capturing the character-to-word representations. The proposed model learns the character-to-word embeddings. Evaluations show that this approach has improved NER, POS, and Chunking compared to the other similar models.

Peters et al. [21] have introduced ELMo, another deep contextualized word representation that utilizes a deep bi-directional language model (biLM) in order to build the word vectors. They evaluated ELMo on six NLP tasks, such as NER.

Devlin et al. [22] have introduced Bidirectional Encoder Representations from Transformers (BERT) word representation. BERT uses both the left and the right contexts of the word. BERT also can be fine-tuned for each task as an additional layer. BERT improved the results in 11natural processing tasks such as NER.

Akbik et al. [23] have proposed Flair embeddings, which leverage a trained character language model to produce acontextual string embedding. In this approach, the same word has different embeddings based on its surrounding text. Like the one proposed by Xin et al. [20], learning in this model is at the character level. The evaluation of the proposed model on the OntoNotes corpus showed state-of-the-art results in NER. Akbik et al. [24] have also introduced a newer development of the Flair embeddings model that presents state-of-the-art results on the WNUT2017 corpus.

Strakova et al. [25] have proposed two neural network architectures (LSTM-CRF architecture), which support named entities with multiple labels (nested NER). They used three contextual embeddings, including ELMo, BERT, and Flair, and reported the results on four different corpora such as CoNLL2003 and multiple languages such as Dutch and Spanish.

Jiang et al. [26] have studied differentiable Neural Architecture Search (NAS) using an RNN-CRF architecture and Flair embedding. Their model outperformed other strong baselines on CoNLL2003 in the NER task. The innovation of this research work was to remove the softmax-local constraint.

Baevski et al. [27] have presented a new approach for pre-training a bi-directional transformer model that uses a cloze-style word reconstruction tasks to make a contextualized word representation. In this architecture, each word is ablated and must be predicted, given the rest of the text. Their approach improved the results of BERT models on different NLP tasks such as NER and constituency parsing, and currently, it is the state-of-the-art model for NER on CoNLL2003.

2.3. NER in the Persian Language

The first notable work on Persian NER is the attempt made by Mortazavi and Shamsfard [28]. Due to the absence of corpora for NER in Persian and the lack of a useful WordNet resource for the Persian language at the time, they proposed a rule-based approach.

In 2010, Rahati et al. [29], introduced an NER system based on a corpus from "Research Center of Intelligent Signal Processing (RCISP)". They used n-grams for feature extraction and four different classifiers, including linear, Bayesian, Nearest neighbor, and Neural network.

In 2014, Kolali Khormuji and Bazrafkan [30] introduced a model based on the local filters to recognize the named entities. In the first step, they used some lookup dictionaries to extract the named entity candidates, and then they filtered false positives from the list. They used the Bijankhan corpus for evaluations, which had four NE types.

In 2014, Moradinasab et al. [31], introduced a rule-based approach to extract the named entities. The system was extracted entities based on the pre-defined patterns. For example, in a sentence started by the "in year" phrase, the system extracts a DATE entity if the token after the phrase is a number.

In another study in 2015, Ahmadi and Moradi [32] presented FarsNERvl, a hybrid method utilizing both the Hidden Markov Model and the rule-based methods combined with three gazetteers to recognize the named entities in Persian, including person names, locations, and organizations. They also introduced their evaluation corpus, a manually annotated collection of different news types from the Mehr News Agency, including 32,606 tokens. They obtained an F1-score of 85.93% on their presented corpus.

In 2016, Poostchi et al. [4] introduced a NER system based on the support-vector machine and hidden Markov model. They also provided ArmanPersoNERCorpus as a manually-annotated Persian NER corpus. They then introduced PersoNER, a NER pipeline for Persian that leverages word embedding and a sequential max-margin classifier. They obtained an F1-score of 75.65% for the person names, 61.59% for the organization names, and 66.67%for the location names on ArmanPersoNERCorpus. They compared their results with two models based on a CRF and a recurrent neural network.

In 2017, Hosseinnejad et al. [33] introduced A'laam corpus, which included 250 tokens and 13 named entity tags. They used a CRF-based model

to train on the new dataset and achieved a precision of 92.94% and recall 78.48% on the introduced dataset.

In another study in 2017, Dashtipour et al. [34] presented a novel scalable system for Persian Named Entity Recognition (PNER). Their proposed method can extract three NER tags, including the person names, locations, and dates. PNER combined a rule-based grammatical approach with machine learning. Their proposed system integrated gazetteers of Persian named entities, Persian grammar rules, and a Support Vector Machine (SVM) model. In order to evaluate their model performance, they constructed a corpus of 1000 articles containing critiques on Iranian and non-Iranian movies collected from two well-known Iranian film websites (caffecinema.com andcinematicket.com), and manually annotated them.

In 2018, Khodakarami [35] created the most extensive NER corpus for the Persian language, including 1,100,00words, three gazetteers, and a NER system that used some various machine-learning methods for extracting the named entities. The system utilized an artificial neural network, a hidden Markov model, and conditional random fields to extract NEs. ANN outperformed the other two methods based on their reports.

In 2018, Poostchi et al. [36] presented a performing approach based on a deep learning architecture (BiLSTM-CRF). They also published several word embeddings for the Persian language. This approach reached 77.45% F1-score on the ArmanPersoNERCorpus. They also publicly released the corpus as the first 6-class NER dataset for the Persian language.

In 2019, the PAYMA [37] corpus, which consisted of 709documents and included 302530 tokens and also an entity recognition software[2], was introduced, but the corpus has not been published yet.

In 2020, Taghizadeh et al. [38] introduced NSURL-2019 Task 7 for named entity recognition in the Persian language and compared seven different models based on the PAYMA corpus containing 300K tokens. The models were tested based on another corpus made by Iran Telecommunication Research Center (ITRC). MorphoBERT, a BiLSTM model that used BERT as its word embedding features, defeated the other algorithms. Beheshti-NER [39] that used Transformer-CRF, and BERT achieved second place in the NSURL-2019competition.

In 2020, Balouchzahi et al. [17] introduced PUNER - Parsi ULMFiT, which used a Transfer Learning (TL) model (based on training a language model on Wikipedia articles) with Universal Language Model Fine Tuning (UML-FiT). The results obtained were compared with a BiLSTM model and five different word embeddings on ArmanPersoNERCorpus and Persian-NER[3] datasets.

In 2020, Momtazi and Torabi [31,40] introduced a corpus including 3000 abstracts from Persian Wikipedia articles and used two different word vector representations (Word2vec and fastText) and a Bidirectional Long Short-Term Memory (BiLSTM) network to extract the named entities.

In 2020, Tafreshi and Soltanzadeh [41] had introduced a CRF model based on syntactic features extracted from the dependency parse tree of the sentences. They evaluated their study using Persian syntactic Dependency Treebank [42]. NER labels are tagged by experts manually. Table 1 summarized the studies and datasets about the named entity recognition in the Persian language.

Table 2 lists all the datasets used in these studies. Originally, two of these datasets are annotated for the POS tagging task. The table defines the publication year (the year in which the dataset was downloadable or introduced), availability, size, and the number of NE tags in each corpus.

2.4. NER on Social Network Data

The recent research works are focused on NER on the social network and noisy data. Some studies, such as [43,44], present the results of the Twitter NER shared task associated with WNUT 2015 and 2016.

Daniken and Cieliebak [45] have used the sentence-level features and transfer learning (that transfer knowledge from a source domain to a target domain) on the Twitter data and have evaluated their work on the WNUT corpus.

Aguilar et al. [46] have presented two BLSTM-CRF systems that address the noisiness of social media texts using three features, including character level phonetics, word embedding, and POS tags[4].

Akbik et al. [24] have proposed a pooled contextualized embeddings model based on the Flair embedding, which addresses the rare strings in the under-specified contexts. They evaluated the methods on CoNLL2003 and WNUT, and the results obtained showed significant improvements

[2] An online demo is available at http://ner.ut.ac.ir

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

[4] Part-of-speech tags

compared to the previous state-of-the-art NER methods.

Table 1. NER studies in the Persian language

Study	Publication year	Method	Dataset
Mortazavi et al. [28]	2009	Rule-based	Mortazavi et al. corpus
Rahati et al. [29]	2010	Linear, Bayesian, Nearest neighbor, ANN	RCISP corpus
Khormuji et al. [30]	2014	Local filters	Hamshahri corpus
Moradinasab et al. [31]	2014	Rule-based	Unknown
Ahmadi and Moradi. [32]	2015	HMM, Rule-based	Mehr News Agency corpus
Poostchi et al. [4]	2016	SVM, HMM	ArmanPersoNERCorpus
Hosseinnejad et al. [33]	2017	CRF	A'laam corpus
Dashtipour et al. [34]	2017	Rule-based, SVM	Dashtipour et al. corpus
KhodaKarami [35]	2018	ANN, HMM, CRF	Khodakarami corpus
Poostchi et al. [36]	2018	BiLSTM-CRF	ArmanPersoNERCorpus
PAYMA [37]	2019	CRF+LSTM+k-means	PAYMA
Taghizadeh et al. [38]	2020	Competition	NSURL-2019 Task 7
Beheshti-NER [39]	2020	BiLSTM-CRF+BERT	NSURL-2019 Task 7
Balouchzahi et al. [17]	2020	TL and UML-FiT	ArmanPersoNERCorpus and Persian-NER
Momtazi and Torabi [40]	2020	BiLSTM+(word2vec/fasttext)	Momtazi and Torabi corpus
Tafreshi and Soltanzadeh [41]	2020	CRF	PerDT [42] (+ NER tags) [41]

Table 2. NER datasets for the Persian language

Name	Publication year	Open dataset	Size	Number of NE tags
Hamshahri (POS-tagging) [47]	2006	yes	2,597,937 tokens	4
Mortazavi et al. corpus [28]	2009	no	Unspecified	40
RCISP corpus (POS-tagging) [29]	2010	no	Approximately 10M tokens	3
Mehr News Agency corpus [32]	2015	no	32606	3
ParsNERCorp	2016	yes	358,831 tokens	4
A'laam corpus	2017	no	250,000 tokens	13
Dashtipour et al. corpus [34]	2017	no	1000 articles about movies	3
ArmanPersoNERCorpus [4]	2018	yes	250,015 tokens	7
KhodaKarami corpus [35]	2018	no	1.1M tokens	4
Persian-NER	2018	yes	25M tokens (not approved manually)	5
ParsNER-Social (this paper)	2019	yes	205,373 tokens	4
PAYMA [37]	2019	yes	302,530 tokens	7
NSURL-2019 Task 7 [38]	2020	yes	300K+600K tokens	4 and 7
Momtazi and Torabi corpus [40]	2020	no	3K Wikipedia articles	16
PerDT (+ NER tags) [41]	2020	no	29982 sentences	3

3. ParsNER-Social

Based on the standard defined in the Conference on Computational Natural Language Learning (CoNLL) in2003 [5], a NER Corpus must consist of words and their NER tags in IOB (Inside-Outside-Beginning) form. The named entities are categorized into four main classes: person (PERS), location (LOC), organization (ORG), and miscellaneous (MISC). The latter refers to the date, percentage, money, and numerical data. In this definition, the date instances are not considered as name entities unless those refer to a specific day in history. For example, the 23rd of July is not a named entity since we have the 23rd of July each year. Each class in the IOB format has a B-X and I-X form (X is a tag name). The B-X form denotes the beginning of a named entity, and the I-X form denotes the inside or end of a

named entity. For example, B-PERS is the beginning of a person's name, and I-PERS is the inside or end of a person's name. If a word is not being named entity, it is tagged as O. Based on these definitions, all the NER corpora in almost all languages can be structured in this format. Figure 1 shows two examples of the NER corpora for two different languages: English on the left side and Persian on the right side. Each word and its tag are always written in one separate line, and a pre-defined delimitator (TAB character in most cases) is used between the word and its tag. Just like the other words, all the punctuation marks are written on separate lines.

with	O	B-PER	محسنی
Del	B-PER	I-PER	اژه‌ای
Bosque	I-PER	O	از
in	O	O	برگزاری
the	O	O	مطلوب
final	O	O	اجلاس
years	O	B-ORG	جنبش
of	O	I-ORG	عدم
the	O	I-ORG	تعهد
seventies	O	O	در
in	O	B-LOC	تهران
Real	B-ORG	O	تقدیر
Madrid	I-ORG	O	کرد
.	O	O	.

Figure 1. Standard English and Persian corpus for NER task.

ParsNERCorp corpus, including ParsNER-News (201460 tokens) and ParsNER-Wiki (157391 tokens) has been publicly released in 2016 in GitHub under the GPL3 license[5]. The corpus contains news data that is originated from Bijankhan Corpus [47]. This article introduces the ParsNER-Social corpus constructed from the social media contents, including 10 Telegram channels in 10 different categories: sport, economics, gaming, IT news, general news, travel, art, academic, fun, and health. The corpus statistics are presented in table 3, showing the number of different NER tags, including the person, location, organization, and miscellaneous named entities in each category.

Figure 2 shows four different sentences from ParsNER-Social. The second column includes part of speech tags, which are automatically calculated and not approved manually. This additional data can work as a feature in the methods.

```
136999  -DOCSTART-  N  0
137000
137001  🌎  N  0
137002  قرارداد  N  0
137003  ذکر  N  0
137004  شده  V  0
137005  قصد  N  0
137006  دارد  V  0
137007  که  CONJ  0
137008  از  P  0
137009  میزان  Ne  0
137010  گرمایش  Ne  0
137011  جهانی  AJe  0
137012  زمین  N  B-LOC
137013  در  P  0
137014  سال  Ne  B-MISC
137015  ۲۱۰۰  NUM  I-MISC
137016  تا  P  0
137017  حدی  N  0
137018  بکاهد  V  0
137019  .  PUNC  0
```

```
136859  ♨  N  0
136860  توضیح  Ne  0
136861  متخصص  AJe  0
136862  چیپ‌های  Ne  0
136863  اپل  N  B-ORG
136864  درباره‌ی  Pe  0
136865  چیپست  N  0
136866  A12X  RES  0
136867  آیپد  RES  0
136868  پرو  N  0
136869  ۲۰۱۸  NUM  B-MISC
```

```
91  🚚  N  0
92  دورهمی  NUM  0
93  کانون  Ne  0
94  کتاب  Ne  0
95  🚚  N  0
96  📅  Ne  0
97  شنبه  N  B-MISC
98  ۲۵  NUMe  I-MISC
99  آذر  N  I-MISC
100  🕐  N  0
101  ساعت  Ne  B-MISC
102  ۱۳  NUM  I-MISC
103  الی  P  I-MISC
104  ۱۵  NUM  I-MISC
105  مجتمع  Ne  B-LOC
106  بهرامی  N  I-LOC
```

```
136400  -DOCSTART-  N  0
136401
136402  📸  N  0
136403  دوربین  Ne  0
136404  فول  AJ  0
136405  فریم  Ne  0
136406  #لایکا  N  0
136407  Q-P  RES  0
136408  با  P  0
136409  قیمت  Ne  0
136410  گزاف  AJe  0
136411  ۴۹۹۵  NUM  B-MISC
136412  دلار  N  I-MISC
136413  معرفی  N  0
136414  شد  V  0
```

Figure 2. Four different sentences from ParsNER-Social.

We used a computer-assisted method in order to construct the ParsNER-Social corpus. In the first step, we selected one Persian Telegram channel in each category manually. By examining several channels in each category, we tried to select the channels so that there was a range of language formality, from informal language to fully formal language, among these channels. The selected Telegram channels have specific topics in many cases, and their editors and writers use entities that are known for their specific audiences, and they mention these entities with their shortest forms. The selected channels in each category are:

–Academic: @IUST_Official
–Art: @academyhonarshamseh
–Economics: @Eghtesadnews_com
–Fun: @khandevaneeh
–Game: @Gamefa_official
–General News: @Akharinkhabar
–Health: @tabbaye_ir
–IT News: @Digiato
–Sports: @varzesh3
–Travel: @AftabeSaheleabi

[5] All ParsNER-Social corpus parts are downloadable at https://github.com/majidasgari/ParsNER/tree/master/persian

Table 3. Number and percentage of each NER tag in ParsNER-Social corpus.

	All	PER	LOC	ORG	MISC	O
Sports	24,139	1,877	505	2632	2,129	16,996
	100.00%	7.78%	2.09%	10.90%	8.82%	70.41%
Economics	28,476	497	509	2,037	1,460	23,973
	100.00%	1.75%	1.79%	7.15%	5.13%	84.19%
Gaming	11,653	49	59	291	768	10,486
	100.00%	0.42%	0.51%	2.50%	6.59%	89.99%
General news	29,404	1,234	711	2,348	1,222	23,889
	100.00%	4.20%	2.42%	7.99%	4.16%	81.24%
IT news	28,442	210	304	1,089	1,623	25,216
	100.00%	0.74%	1.07%	3.83%	5.71%	88.66%
Travel	26,294	33	1,781	686	3,505	20,289
	100.00%	0.13%	6.77%	2.61%	13.33%	77.16%
Art	13,848	1,093	300	204	460	11,791
	100.00%	7.89%	2.17%	1.47%	3.32%	85.15%
Academic	17,386	492	912	1,807	1,698	12,477
	100.00%	2.83%	5.25%	10.39%	9.77%	71.76%
Fun	19,019	1,814	200	589	1,006	15,410
	100.00%	9.54%	1.05%	3.10%	5.29%	81.02%
Health	6,712	305	87	39	342	5,939
	100.00%	4.54%	1.30%	0.58%	5.10%	88.48%
Total	205,373	7,604	5,368	11,722	14,213	166,466
	100.00%	3.70%	2.61%	5.71%	6.92%	81.06%

Then we implemented a crawler software to fetch up to 40000 tokens from each channel via Telegram API. The crawler was launched in January 2019 and collected the required raw data. Many of the Telegram channels in Persian contains duplicated advertisement posts. That is why we implemented other software to remove such posts automatically. Afterward, we choose the most popular posts based on their view counts. Finally, 10822 documents were collected, including 20025 sentences and 205373 tokens, and an average of 10.25 tokens per sentence.

Figure 3 shows a comparison between the ParsNER-Corp corpus and its subsets, including ParsNER-News and ParsNER-Wiki, and ParsNER-Social corpus. As can be seen, the number of tokens per sentence is much higher in ParsNERCorp than in ParsNER-Social due to the different nature of news texts in the official language and texts available on social media in the informal language.

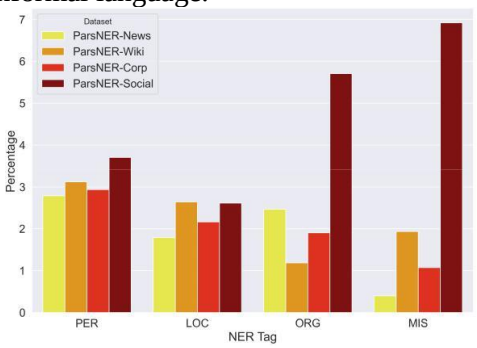

Figure 3. Percentage of NER Tags among different ParsNER corpora.

Table 4 also shows each NER tag's ratio to the total number of tokens in each corpus. As can be seen, the ratio of NER tags in the ParsNER-Social corpus is much higher than in other corpora, which is further evidence that texts on social media are very different from other types of texts and contain more named entities.

Table 4. Comparison between the ParsNERCorp corpus subsets and ParsNER-Social corpus.

Dataset	Tokens	Sentences	Token/Sentence	PER	LOC	ORG	MISC	O
ParsNER-News	201,460	6,655	30.27	5,620	3,610	4,969	803	179,773
ParsNER-Wiki	157,391	6,971	22.58	4,914	4,156	1,865	3,048	136,414
ParsNER-Corp	358,851	13,626	26.34	10,534	7,766	6,834	3,851	316,187
ParsNER-Social	205,373	20,025	10.25	7,604	5,368	11,722	14,213	166,466

In the next step, we automatically annotated all the selected posts by ParsNER, a baseline NER method (described in the next section).

Finally, three human experts in NER tagging, who were fluent in Persian, were asked to check the ParsNER automatically annotated output and fix

the possible errors. The corpus was distributed among two out of three experts so that each NER tag annotated by the machine was checked and approved by at least two out of three experts. In other words, each NER tag was verified by at least one expert, and at least one expert double-checked this verification. In situations where there was disagreement between the two experts, the third expert, who had more knowledge of NER tags, judged between them and approved one of the two cases. For example, in a sentence with a phone number in it, the first expert tagged it as O, and the second expert tagged it as MISC. In this case, the third expert-approved the annotation by the second expert. In another example, in a sentence in which the word "television" was used to refer to Iranian state television, one of the experts tagged it as ORG, and the other tagged it as O. In this case, the third expert confirmed the opinion of the first expert.

4. Experiments

This section presents the results obtained from two baselines and one state-of-the-art deep learning method with six different configurations on our newly published dataset, ParsNER-Social. We carried out 3-fold cross-validation experiments to evaluate the baseline methods and 5-fold cross-validation for the deep learning method. The folds are publicly available for further experiments. Table 5 shows the results of Stanford NER and ParsNER as the baseline methods on the ParsNER-Social corpus.

4.1. Experiment Setup and Models

4.1.1. Baseline 1

We used the original Stanford NER software[6] that implemented the conditional random fields and analyses input text and recognized the named entities after some regular preprocessing steps. No feature nor gazetteer, specially designed for the Persian language, was not added to this baseline method (wordList was a standard feature for Stanford NER and was included in the baseline method with a list of words in the Persian language).

4.1.2. Baseline 2

In order to automatically annotate ParsNER-Social and possession a baseline method specially designed for the Persian language, we implemented ParsNER[7], which is a system based on the Stanford NER tagger. In the training, testing, and prediction phases, it uses normalization, sentence tokenizing, word tokenizing, part of speech tagging, and dependency parsing as the pre-processing steps. ParsNER also uses JHazm in the pre-processing pipeline. JHazm[8] is a Java implementation for Hazm library[9] that is written in the Python programming language. ParsNER uses the following features: language modeling features, POS tags, wordlist, postfix and prefix, keyword lists, name gazetteers, NE lists, dependency parsing, and Wikipedia infoboxes. ParsNER has a post-tagging phase that improves the recall for Person names.

4.1.3. Deep Learning Method

As mentioned in Section 2, in recent years, the deep learning models have overcome other NER methods. We evaluated multiple deep learning approaches and configurations using our new dataset. Many state-of-the-art approaches use an advanced language model in order to increase the learning accuracy of the models. For this purpose, instead of using the word vectors, transformers have been used. BERT is a bidirectional transformer that is pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising Toronto Book Corpus and Wikipedia.

In this paper, we implemented six different network configurations, which were trained on the new dataset:

- Bi-LSTM. The first configuration uses another combination of layers, namely a Bi-LSTM layer and a linear classification layer.
- LSTM+CRF. LSTM layers are very suitable for the NER task since they learn the relationship of the words in a sentence. The second network configuration includes an LSTM layer and a CRF layer. Afterward, a layer is used for the classification. This model has a lower accuracy than the other configurations.
- Bi-LSTM+Linear. In order to examine the effect of the number of the dense linear layer on the results, we added another dense layer to the first network. This new layer was placed between the Bi-LSTM's output and the classification layer. This

[6] https://nlp.stanford.edu/software/CRF-NER.shtml
[7] https://github.com/majidasgari/ParsNER
[8] https://github.com/mojtaba-khallash/JHazm
[9] https://github.com/sobhe/hazm

setting led to better results compared to the first configuration.

- Linear. This configuration uses two dense layers with768 nodes, where one of them is used for classification. This network setting achieved a significant improvement in the F1-score in our experiments on ParsNER-Social.
- BERT-TokenClassification. In this configuration, we use the BERT-TokenClassification model without any other layers, a BERT transformer with a linear layer, and a dropout layer. This configuration outperforms the previous configurations.

4.1.4. BERT-Token Classification with ParsBERT

Since the BERT-based models are usually focused on English, there are two approaches to use them in other languages. The first approach for using BERT in other languages is to use the Multilingual BERT (MLBERT), which is constructed with limited resources. In the above experiments, we used the same model. The second approach constructs monolingual BERT. ParsBERT [48] has recently been introduced as a monolingual BERT, especially for the Persian language, which is pre-trained over a massive Persian language dataset. While in the above experiments, BERT-TokenClass shows the best performance among the other configurations, this time, we trained the BERT-TokenClass model with ParsBERT. The experiment showed a significant improvement in the results of NER on ParsNER-Social. In order to prepare the data for entering the network, we used a PyTorch Dataset module, in which additional characters were removed, and using the ParsBert Tokenizer, we tokenized the sentences and also more preprocessing tasks to clean the data. Depending on the length of the sentences, we used 128 tokens for padding. The data was divided into two partitions, 80% for training, and 20% for testing. Also, 5-fold cross-validation was used during the training of the network. We used different types of BERT transformers to train the network. ParsBERT transformer with 768 features had the highest accuracy. The BERT-TokenClassification module uses these 768 features to feed a linear layer with the output of the desired number of classes. In order to optimize the network, we used Adam optimizer with a learning rate of 0.0001 and the CrossEntropyLoss function to calculate the loss of the network. Also, the batch size was set to 128. Each iteration took about 25 minutes.

4.2. Discussion

We evaluated different models on ParsNER-Social, the results of which are shown in table 5. Regarding the two baseline methods, since they used relatively older approaches, as expected, they performed lower than the other method. As expected, our baseline model, ParsNER, performed better than the Stanford NER model because it used the Persian-specific features.

Looking at the evaluation results, the BERT-TokenClassification model offers better results compared to the other models and configurations. When this model uses ParsBERT instead of MLBERT, the results show a significant improvement. The evaluations showed that the other models that were examined, although they were conventional and advanced models, could not achieve the BERT-based models' performance. The simplest model, linear, showed the best performance among the BERT-based models, which could be a significant point for further research works.

5. Release of the Resources

All the resources obtained as a result of this work are freely downloadable and available to the research community at https://github.com/majidasgari/ParsNER. Among these resources; we included ParsNER-Social corpus. The corpus is also stored separately by category and in different folds (3-folds) in this repository.

6. Conclusions and Future Works

In this paper, we introduced ParsNER-Social, a public CoNLL-standard corpus for Persian Named Entity Recognition (NER) in social media texts, an essential requirement for machine learning approaches to NER in Persian and fitting in general-domain settings.

The ParsNER-Social corpus is published as a continuation of our previously published ParsNERCorp corpus. Various comparisons between these two corpora's specifications were presented in this paper, and it was observed that the ratio of the number of tokens per sentence in ParsNER-Social was less than ParsNERCorp. It was also observed that the number of NER tags in ParsNER-Social was more than ParsNERCorp. These differences are due to social media texts' nature compared to the other texts and their less formal language and were evidence that ParsNER-Social was more suitable for more general applications.

We also trained and evaluated three supervised methods, including one state-of-the-art deep learning model with six different configurations on the ParsNER-Social corpus. Our experiments showed that the Bidirectional Encoder Representations from Transformers (BERT)-based models outperformed the other approaches on the ParsNER-Social corpus.

For future works based on this research work, we are considering a variety of cases. Most importantly, in the future, we plan to increase the size of the ParsNER-Social corpus and use more diverse domains from the Telegram channels and augment the dataset by adding some content from other popular social networks among Persian speakers such as Twitter. We also plan to introduce newer NER models with improved performance. We have used two pre-trained BERT models, Multilingual BERT and ParsBERT, a monolingual BERT for the Persian language. The BERT model can be improved by increasing the number of tokens and using more texts originated from various domains. A more fine-tuned BERT for NER on texts extracted from Persian social media can significantly improve the NER results.

Moreover, we are working on building a monolingual BERT language model, namely FarsBERT, built for the Persian language from scratch. Utilizing FarsBERT and fine-tuning it for NER on ParsNER-Social also improves the NER results in Persian. We have also considered creating a participatory framework to get help from the users of social networks such as Twitter and Telegram to increase the volume of the ParsNER-Social corpus.

Acknowledgment

It is necessary to acknowledge the active collaboration of Dr. Sayyed Ali Hossayni, who kindly collaborated with us during the conduction of this research.

References

[1] M. E. Khademi and M. Fakhredanesh, “Persian automatic text summarization based on named entity recognition”. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pp. 1–12, 2020.

[2] R. Grishman and B. Sundheim. “Message understanding conference- 6: A brief history”. *In proceedings of the 16th international conference on computational linguistics*, COLING, 1996, pp. 466–471.

[3] A. Borthwick and R. Grishman, “A maximum entropy approach to named entity recognition”, Ph.D. dissertation, New York university, 1999.

[4] H. Poostchi, E. Z. Borzeshi, Abdous, M., and M. Piccardi, “PersoNER: Persian named-entity recognition”. *In proceedings of the 26th international conference on computational linguistics, proceedings of the conference: Technical papers*, COLING 2016, 2016, pp. 3381–3389.

[5] E. F. T. K. Sang and F. D. Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition”. *In Proceedings of the seventh conference on natural language learning*, CoNLL-2003, held in cooperation with HLT-NAACL, 2003, pp. 142–147.

[6] A. Farzindar and D. Inkpen. “Natural language processing for social media”. *Synthesis Lectures on Human Language Technologies*, vol 8(2), pp. 1–166, 2015.

[7] Y. Kim, J. Kim, and J. Seo, “Noise improves noise: Verification of pre-training effect with weakly labelled data on social media NER”. *In 2020 IEEE international conference on big data and smart computing*, BigComp, 2020, pp. 225–228.

[8] R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, and N. Xue, “Ontonotes: A large training corpus for enhanced processing”. *Handbook of Natural Language Processing and Machine Translation*. Springer, 2011, pp. 59–66.

[9] L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham, “Results of the wnut2017 shared task on novel and emerging entity recognition”. *In Proceedings of the 3rd workshop on noisy user-generated text*, 2017, pp. 140–147.

[10] J. Li, A. Sun, J. Han, and C. Li, “A survey on deep learning for named entity recognition”. *IEEE Transactions on Knowledge and data Engineering*, pp. 1–20, 2020.

[11] F. Saad, H. Aras, and R. Hackl-Sommer, “Improving named entity recognition for biomedical and patent data using BiLSTM deep neural network models”. *International conference on applications of natural language to information systems*, 2020, pp. 25–36.

[12] Zhou, C., Li, B., & Sun, X. “Improving software bug-specific named entity recognition with deep neural network”. *Journal of Systems and Software*, 2020.

[13] R. Sharma, S. Morwal, B. Agarwal, R. Chandra, and M. S. Khan, “A deep neural network-based model for named entity recognition for Hindi language”. *Neural Computing and Applications*, pp. 1–13, 2020.

[14] V. Yadav and S. Bethard, “A survey on recent advances in named entity recognition from deep learning models”. *In proceedings of the 27th*

international conference on computational linguistics, COLING 2018, 2018, pp. 2145–2158.

[15] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)". *In proceedings of the 7th international workshop on semantic evaluation*, SEMEVAL@NAACL-HLT 2013, 2013, pp. 341–350.

[16] S. Zhang, and N. Elhadad, "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts". *Journal of Biomedical Informatics*, vol. 46(6), pp. 1088–1098, 2013.

[17] F. Balouchzahi and H. Shashirekha, "Puner - Parsi ULMFiT for named-entity recognition in Persian texts". *EasyChair Preprint*, no.4224, 2020.

[18] S. Liu, B. Tang, Q. Chen, and X. Wang, "Effects of semantic features on machine learning-based drug name recognition systems: Word embeddings vs. manually constructed dictionaries". *Information*, vol. 6(4), pp. 848–865, 2015.

[19] M. Habibi, L. Weber, M. L. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition". *Bioinformatics*, vol. 33(14), pp. 37–48, 2017.

[20] Y. Xin, E. Hart, V. Mahajan, and J. D. Ruvini, "Learning better internal structure of words for sequence labelling". *In proceedings of the 2018 conference on empirical methods in natural language processing*, EMNLP 2018, 2018, pp. 2584–2593.

[21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations". *In proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies*, vol. 1 (long papers) 2018, pp. 2227–2237.

[22] J. Devlin, M. Chang, L. Kristina, and K. Toutanova, "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*, NAACL-HLT 2019, vol. 1 (long and short papers), 2019, pp. 4171–4186.

[23] A. Akbik, D. Blythe, and R. Vollgraf. "Contextual string embeddings for sequence labelling". *In proceedings of the 27th international conference on computational linguistics*, 2018, pp. 1638–1649.

[24] A. Akbik, T. Bergmann, and R. Vollgraf, "Pooled contextualized embeddings for named entity recognition". *In proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies*, vol. 1 (long and short papers), 2019, pp. 724–728.

[25] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization". *In proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 5326–5331.

[26] Y. Jiang, C. Hu, T. Xiao, C. Zhang, and J. Zhu, "Improved differentiable architecture search for language modelling and named entity recognition". *In proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing*, EMNLP-IJCNLP, 2019, pp. 3585–3590.

[27] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven pre-training of self-attention networks". *In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing*, EMNLP-IJCNLP, 2019, pp. 5359–5368.

[28] P. S. Mortazavi and M. Shamsfard. "Named entity recognition in Persian texts". *In proceedings of the 15th national computer society of Iran conference*, 2009, pp. 1–10.

[29] S. Rahati-Ghoochani, S. A. Esfahani, and J. Nader, "Persian name entity recognition and classification". *Signal and Data Processing*, 2010.

[30] M. Kolali Khormuji and M. Bazrafkan, "Persian named entity recognition based with local filters". *International Journal of computer Applications*, vol. 100(4), 2014.

[31] O. Moradiannasab, S. Momtazi, and A. Palmer, "A named entity recognition tool for Persian". *In proceedings of the 3rd Iranian conference on computational linguistics*, 2014.

[32] F. Ahmadi and H. Moradi, "A hybrid method for Persian named entity recognition". *In 2015 7th conference on information and knowledge technology (IKT)*, 2015, pp. 1–7.

[33] S. Hosseinnejad, Y. Shekofteh, and T. Emami Azadi, "A'laam corpus: A standard corpus of named entity for Persian language". *Signal and Data Processing*, vol. 14(3), 2017, pp. 127–142.

[34] K. Dashtipour, M. Gogate, A. Adeel, A. Algarafi, N. Howard, and A. Hussain. "Persian named entity recognition". *In 2017 IEEE 16th international conference on cognitive informatics and cognitive computing*, ICCI* CC, 2017, pp. 79–83.

[35] M. Khodakarami, "Toward implementation of a named entity recognition system using machine learning methods", Ph.D. dissertation, University of Tehran, 2018.

[36] H. Poostchi, E. Z. Borzeshi, and M. Piccardi, "BiLSTM-CRF for Persian named-entity recognition ArmanPersoNER corpus: the first entity-annotated Persian dataset". *In proceedings of the eleventh*

international conference on language resources and evaluation, LREC 2018, 2018, pp. 4427–4431.

[37] M. S. Shahshahani, M. Mohseni, A. Shakery, and H. Faili, "Payma: A tagged corpus of Persian named entities". *Signal and data processing*, vol. 16(1), 2019.

[38] N. Taghizadeh, Z. Borhanifard, M. GolestaniPour, and H. Faili. "NSURL-2019 task 7: Named entity recognition (NER) in Farsi". *arXiv preprint arXiv:2003.09029*, 2020.

[39] E. Taher, S. A. Hoseini, and M. Shamsfard, "Beheshti-NER: Persian named entity recognition using BERT". *arXiv preprintarXiv:2003.08875*, 2020.

[40] S. Momtazi and F. Torabi, "Named entity recognition in Persian text using deep learning". *Signal and Data Processing*, vol. 16(4), pp. 93–112, 2020.

[41] L. Jafar Tafreshi and F. Soltanzadeh. "A novel approach to conditional random field-based named entity recognition using Persian specific features". *Journal of AI and Data Mining*, vol. 8(2), pp. 227–236, 2020.

[42] M. S. Rasooli, M. Kouhestani, and A. Moloodi, "Development of a Persian syntactic dependency treebank". *In Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 306–314.

[43] T. Baldwin, M. de Marneffe, B. Han, Y. Kim, A. Ritter, and W. Xu, "Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition". *In proceedings of the workshop on noisy user-generated text*, NUT@IJCNLP 2015, 2015, pp. 126–135.

[44] B. Strauss, B. Toma, A. Ritter, M. de Marneffe, and W. Xu, "Results of the WNUT16 named entity recognition shared task". *In proceedings of the 2nd workshop on noisy user-generated text*, NUT@COLING 2016, 2016, pp. 138–144.

[45] P. von Daniken and M. Cieliebak, "Transfer learning and sentence-level features for named entity recognition on tweets". *In proceedings of the 3rd workshop on noisy user-generated text* (pp.166–171). Association for Computational Linguistics, 2017.

[46] G. Aguilar, A. Pastor´Lopez-Monroy, F. A. Gon´zalez and T. Solorio "Modeling noisiness to recognize named entities using multitask neural networks on social media". *In proceeding of 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies*, NAACL HLT 2018, 2018, pp. 1401–1412.

[47] F. Oroumchian, S. Tasharofi, H. Amiri, H. Hojjat, and F. Raja, "Creating a feasible corpus for Persian pos tagging" (UOWD Technical Reports Series No. no. TR 3/2006). Dubai Campus: University of Wollongong, 2006.

[48] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-based model for Persian language understanding". *arXiv preprint arXiv:2005.12515*, 2020.

ParsNER-Social: یک پیکره متنی برای شناسایی موجودیت‌های نامدار در متون رسانه‌های اجتماعی فارسی

مجید عسگری بیدهندی، بهروز جانفدا، امیدرضا روشنی‌طلب و بهروز مینایی- بیدگلی*

دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.

ارسال ۲۰۲۰/۰۹/۱۰؛ بازنگری ۲۰۲۰/۱۰/۱۲؛ پذیرش ۲۰۲۰/۱۱/۲۵

چکیده:

شناسایی موجودیت‌های نام‌دار یکی از پیش‌نیازهای اساسی بسیاری از کارهای پردازش زبان طبیعی است. کلیه پیکره‌های متنی عمومی برای شناسایی موجودیت‌های نامدار در فارسی مانند ParsNERCorp و ArmanPersoNERCorpus، بر اساس متون خبری مانند پیکره بی‌جن‌خان مربوط به سال ۲۰۰۴ ساخته شده‌اند. به همین ترتیب، بیشتر مدلهای شناسایی موجودیت نامدار در فارسی برای کار روی متون خبری تنظیم می‌شوند و به اندازه کافی برای به کار گرفته شدن روی متونی با محتوای غیرخبری، مانند متون شبکه‌های اجتماعی، انعطاف پذیر نیستند. این مطالعه پیکره‌ی ParsNER-Social را معرفی می‌کند. این پیکره برای آموزش و ساخت مدل‌های شناسایی موجودیت نام‌دار از متون شبکه‌های اجتماعی فارسی مناسب است. این پیکره‌ی متنی از ۲۰۵۳۷۳ نشانه و برچسب آنها تشکیل شده است که از محتوای شبکه های اجتماعی شامل ۱۰ کانال تلگرام در ۱۰ موضوع مختلف جمع شده‌اند. علاوه بر این، سه روش باناظر بر اساس مجموعه ParsNER-Social معرفی شده و آموزش داده شده‌اند: دو مدل مبتنی بر میدان‌های تصادفی شرطی به عنوان مدل‌های پایه، و یک مدل یادگیری عمیق پیشرفته با شش پیکربندی مختلف. این مدل‌ها روی مجموعه داده پیشنهادی ارزیابی شده است. آزمایشات نشان می‌دهد که مدل‌های فارسی تک زبانه مبتنی بر بازنمایی رمزگذار دو جهته از ترانسفورماتورها (MLBERT) از سایر رویکردها در مجموعه ParsNER-Social پیشی گرفته است. در میان پیکربندیهای مختلف مدلهای MLBERT، مدل ParsBERT + BERT-TokenClass نمره ۸۹/۶۵ در معیار F1 بدست آورد..

کلمات کلیدی: تشخیص موجودیت‌های نامدار، پردازش زبان طبیعی، پیکره متنی رسانه‌های اجتماعی، زبان فارسی.