PersoNER: Persian Named-Entity Recognition

Hanieh Poostchi
University of Technology Sydney
Capital Markets CRC
hpoostchi@cmcrc.com

Ehsan Zare Borzeshi
Capital Markets CRC
ezborzeshi@cmcrc.com

Mohammad Abdous
Iran University of Science and Technology
md.abdous@gmail.com

Massimo Piccardi
University of Technology Sydney
massimo.piccardi@uts.edu.au

Abstract

Named-Entity Recognition (NER) is still a challenging task for languages with low digital resources. The main difficulties arise from the scarcity of annotated corpora and the consequent problematic training of an effective NER pipeline. To abridge this gap, in this paper we target the Persian language that is spoken by a population of over a hundred million people world-wide. We first present and provide ArmanPerosNERCorpus, the first manually-annotated Persian NER corpus. Then, we introduce PersoNER, an NER pipeline for Persian that leverages a word embedding and a sequential max-margin classifier. The experimental results show that the proposed approach is capable of achieving interesting MUC7 and CoNNL scores while outperforming two alternatives based on a CRF and a recurrent neural network.

1 Introduction

Named-Entity Recognition (NER), introduced in the sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996), concerns the recognition of Named Entities (NE) and numeric expressions in unstructured text. Since 1996, great effort has been devoted to NER as a foundational task for higher-level natural language processing tasks such as summarization, question answering and machine translation.

Shortage of gold standards has initially limited NER investigation to high-resource languages such as English, German and Spanish (Tjong Kim Sang and De Meulder, 2003). Gradually, publicly available encyclopediae have enabled combinations of semi-supervised and distant supervision approaches for other languages (Althobaiti et al., 2015). However, low-resource languages still face a significant scarcity of public repositories. For instance, only 8.8% of Wikipedia articles in Hindi are identified as entity-based articles in Freebase (Al-Rfou et al., 2015). In this work, we aim to enable supervised NER for a low-resource language, namely Persian, by providing the first manually-annotated Persian NE dataset. The Persian language, despite accounting for more than a hundred million speakers around the globe, has been rarely studied for NER (Khormuji and Bazrafkan, 2014) and even text processing (Shamsfard, 2011). In addition, we present PersoNER, a Persian NER pipeline consisting of a word embedding module and a sequential classifier based on the structural support vector machine (Tsochantaridis et al., 2005). The proposed pipeline achieves interesting MUC7 and CoNNL scores and outperforms two alternatives based on a CRF and a recurrent neural network.

2 Related Work

Early research on NER was mostly devoted to handcrafted rule-based systems which are intrinsically language-dependent, and thus laborious to be extended to new languages. As a consequence, recent studies are mainly focused on language-independent machine learning techniques that attempt to learn

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3381–3389, Osaka, Japan, December 11-17 2016.

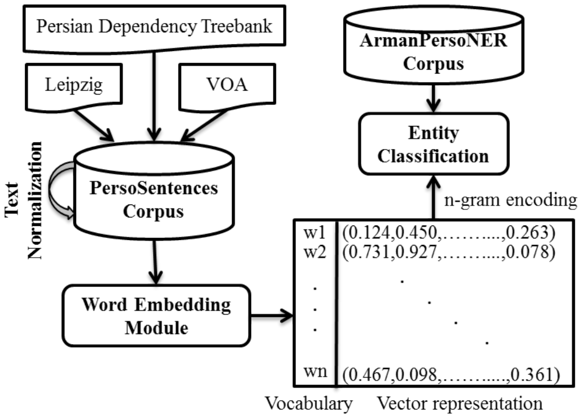


Figure 1: PersoNER workflow.

statistical models for NER from data (Nadeau and Sekine, 2007). Moreover, replacement of manually-annotated gold standards with very large "silver standard" corpora mollifies the scarcity of supervised data. Silver standards are NE annotated corpora derived from processing Wikipedia's text and meta-information alongside entity databases such as Freebase (Nothman et al., 2013; Al-Rfou et al., 2015).

Existing NER approaches mainly divide over two categories: in the first, the task is decoupled into an initial step of word embedding, where words are mapped to feature vectors, followed by a step of word/sentence-level classification. The feature vector can be as simple as a binary vector of text features like '*word is all uppercased*' or a more complex, real-valued vector capturing semantic and syntactic aspects of the word. Word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and Hellinger-PCA (Lebret and Collobert, 2014) are well-known examples of unsupervised word embeddings applied successfully to the NER task. For classification, sequential classifiers such as HMMs (Zhou and Su, 2002), CRFs (Lafferty et al., 2001; Finkel et al., 2005) and deep neural networks (Al-Rfou et al., 2015) have been amongst the most popular choices.

The second category, proposed by (Collobert et al., 2011) and recently followed by many including (Mesnil et al., 2013; Mesnil et al., 2015) and others, leverages recurrent neural networks (RNNs) to deliver end-to-end systems for NER. With this approach, an implicit word embedding is automatically extracted in the network's early layers by initializing the training with random values or a preliminary embedding. In this paper, we apply and compare approaches from both categories.

3 The Proposed Approach

The workflow of PersoNER is illustrated in Figure 1. The steps include data collection, text normalization, word embedding and entity classification. In this section, we focus on the two technical modules, word embedding and classification, while data collection and text normalization are described in Section 4.

3.1 Word Embedding

Term-frequency (tf), term-frequency inverse-document-frequency (tf-idf), bag of words (bow) and word co-occurrence are general statistics intended to characterize words in a collection of documents. Out of them, word co-occurrence statistics have the ability to represent a word by the frequencies of its surrounding words which well aligns with the requirements of NER. Recently, Lebret and Collobert (2014) have shown that a simple spectral method analogous to PCA can produce word embeddings as useful as those of neural learning algorithms such as word2vec. Given an unsupervised training corpus and a vocabulary, V, the co-occurrence matrix, $C_{|V|\times|D|}$, in (Lebret and Collobert, 2014) is computed as:

$$C(v_i, d_j) = p(d_j|v_i) = \frac{n(v_i, d_j)}{\sum_d n(v_i, d)} \tag{1}$$

where $v_i \in V; i = 1 \dots |V|$ and $d_j \in D \subseteq V; j = 1 \dots |D|$. $n(v_i, d_j)$ is the count of occurrences of context word d_j in the neighborhood of reference word v_i. Thus, $C(v_i, :)$ represents discrete probability distribution $p(d|v_i)$ and is used to characterize v_i. Since words are represented as discrete distributions, Lebret and Collobert (2014) argue that it is more appropriate to measure their distances in a Hellinger space. Accordingly, $H(C)$ is the transformation of C into Hellinger space where the distance between any two discrete probability distributions, P and Q, is given by:

$$dist(P, Q) = \frac{1}{\sqrt{2}} ||\sqrt{P} - \sqrt{Q}||_2. \tag{2}$$

Eventually, PCA is applied to reduce the dimensionality of $H(C) \in \mathbb{R}^{|V| \times |D|}$ to $h(C) \in \mathbb{R}^{|V| \times m}$, where $m \ll |D|$.

3.2 Classification

In this subsection, we first briefly introduce sequential labeling as a formal problem and then describe the sequential classifier based on the structural support vector machine.

3.2.1 Sequential Labeling

Sequential labeling predicts a sequence of class labels, $y = \{y_1, \dots y_t, \dots y_T\}$, based on a corresponding sequence of measurements, $x = \{x_1, \dots x_t, \dots x_T\}$. It is a very common task in NLP for applications such as chunking, POS tagging, slot-filling and NER. A widespread model for sequential labeling is the hidden Markov model (HMM) that factorizes the joint probability of the measurements and the labels, $p(x, y)$, by arranging the latter in a Markov chain (of order one or above) and conditioning the measurement at frame t on only the corresponding label. For an HMM of order one, $p(x, y)$ is expressed as:

$$p(x, y) = p(y_1) \prod_{t=2}^{T} p(y_t|y_{t-1}) \prod_{t=1}^{T} p(x_t|y_t) \tag{3}$$

where $p(y_1)$ is the probability of the initial class, terms $p(y_t|y_{t-1})$ are the transition probabilities and terms $p(x_t|y_t)$ are the emission, or measurement, probabilities. By restricting the emission probabilities to the exponential family, i.e., $p(x_t|y_t) \propto exp(w^T f(x_t, y_t))$, the logarithm of probability $p(x, y)$ can be expressed as the score of a *generalized linear model*:

$$\begin{gathered} \ln p(x, y) \propto w^T \phi(x, y) = \\ w_{in} f(y_1) + \sum_{t=2}^{T} w_{tr}^T f(y_t, y_{t-1}) + \sum_{t=1}^{T} w_{em}^T f(x_t, y_t) \end{gathered} \tag{4}$$

where w_{in}, w_{tr} and w_{em} are the linear models for assigning a score to the initial classes, transitions and emissions, respectively. Functions $f(y_1)$, $f(y_t, y_{t-1})$ and $f(x_t, y_t)$ are arbitrary, fixed "feature" functions of the measurements and the labels.

The generalized linear model in (4) is more suitable for discriminative training than the generative probabilistic model in (3). Notable discriminative approaches are conditional random fields (CRFs) (Lafferty et al., 2001) and structural SVM (Tsochantaridis et al., 2005). In particular, structural SVM has built a very strong reputation for experimental accuracy in NLP tasks (Joachims et al., 2009; Tang et al., 2013; Qu et al., 2014) and for this reason we exploit it in our NER pipeline.

Eventually, given a measurement sequence x in input, inference of the optimal label sequence can be obtained as:

$$\bar{y} = \underset{y}{\operatorname{argmax}}\, p(x, y) = \underset{y}{\operatorname{argmax}}(w^T \phi(x, y)) \tag{5}$$

This problem can be efficiently solved in $O(T)$ time by the Viterbi algorithm working in either the linear or logarithmic scale (Rabiner, 1989).

3.2.2 Structural SVM

From a supervised training set of sequences, $\{X, Y\} = \{x^i, y^i\}, i = 1 \dots N$, *structural SVM* finds the model's parameters, w, by minimizing the usual SVM trade-off between the hinge loss and an $L2$ regularizer (Tsochantaridis et al., 2005). Its learning objective can be expressed as:

$$\begin{aligned}
&\underset{w,\xi}{\operatorname{argmin}} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi^i \quad s.t. \\
&w^T\phi(x^i, y^i) - w^T\phi(x^i, y) \geq \Delta(y^i, y) - \xi^i, \\
&\quad i = 1 \dots N, \ \forall y \in \mathcal{Y}
\end{aligned} \tag{6}$$

In the objective function, the first term is the regularizer while the second term, $\sum_{i=1}^{N}\xi^i$, is the hinge loss, i.e. a convex upper bound over the total loss on the training set. Hyperparameter C is an arbitrary, positive coefficient that balances these two terms. In the constraints, $w^T\phi(x, y)$ computes the generalized linear score for a (x, y) pair. In the case of sequential labeling, such a score is given by Eq. (4). Eventually, $\Delta(y^i, y)$ is the loss function chosen to assess the loss over the training set.

For an NER task with M entity classes, each sequence of length T adds $(M + 1)^T$ constraints to (6). Due to their exponential number, exhaustive satisfaction of all constraints is infeasible. However, (Tsochantaridis et al., 2005) has shown that it is possible to find ϵ-correct solutions with a subset of the constraints of polynomial size consisting of only the "most violated" constraint for each sequence, i.e. the labeling with the highest sum of score and loss:

$$\begin{aligned}
&\xi^i = \max_y(-w^T\phi(x^i, y^i) + w^T\phi(x^i, y) + \Delta(y^i, y)) \\
&\rightarrow \bar{y}^i = \underset{y}{\operatorname{argmax}}(w^T\phi(x^i, y) + \Delta(y^i, y))
\end{aligned} \tag{7}$$

This problem is commonly referred to as "loss-augmented inference" given its resemblance with the common inference of Eq. (5) and is the core of structural SVM. In the case of scores and losses that can be computed frame by frame (such as the 0-1 loss or the Hamming loss), the Viterbi algorithm with appropriate weights can still be used to compute the loss-augmented inference in $O(T)$ time.

4 Data Collection

In this section, we describe the collection and preprocessing of the Persian corpora. The datasets consist of 1) an unsupervised corpus, called PersoSentencesCorpus, that we use for the word embedding module and 2) a manually named-entity annotated data set of Persian sentences, called ArmanPersoNERCorpus, that we use for supervised classification. Alongside this publication, we release ArmanPersoNERCorpus[1] as the first ever publicly-available Persian NER dataset.

4.1 PersoSentencesCorpus

A very large corpus of documents covering a variety of contexts is required to populate an effective co-occurrence matrix. We fulfill this requirement by accumulating the following three datasets of Persian sentences:

- The *Leipzig corpora* [2] with 1,000,000 sentences from news crawling and 300,000 from Wikipedia.
- The *VOA*[3] news dataset with 277,000 sentences.
- The *Persian Dependency Treebank*[4] with 29,982 sentences (Rasooli et al., 2013).

The aggregated corpus, called PersoSentencesCorpus, holds more than 1.6 million sentences and seems of adequate size to train the co-occurrence matrix.

[1] http://poostchi.info/hanieh/NLP/ArmanPersoNERCorpus.txt

[2] http://corpora2.informatik.uni-leipzig.de/download.html

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

[4] http://dadegan.ir/en/perdt/

Entity type	**Person**	**Organization**	**Location**	**Facility**	**Event**	**Product**	**Other**
Number of Tokens (NT)	5,215	10,036	4,308	1,485	2,518	1,463	224,990
Percentage	2.08%	4.01%	1.72%	0.59%	1.00%	0.58%	89.99%
Number of Unique-Tokens (NUT)	1,829	1,290	832	548	556	634	15,677
Percentage (NUT/NT)	35.07%	12.85%	19.31%	36.90%	22.08%	43.33%	6.96%

Table 1: Class percentages in ArmanPersoNERCorpus.

4.2 ArmanPersoNERCorpus

To create an NE dataset, in collaboration with ArmanSoft[5], we have decided to manually annotate NEs in a subset of the **BijanKhan**[6] (Bijankhan et al., 2011) corpus which is the most-established tagged Persian corpus, yet lacking entity annotation. We selected the subset from news sentences since they are the most entity-rich. Before the annotation, a comprehensive manual was designed based on the definition of Sekine's extended named entities (Sekine, 2007) adapted to the Persian Language. The annotation task was led by an experienced lead annotator who instructed the front-end annotators (two native post-graduate students) and revised their annotations. The guidelines were very clear and we expected minimal subjectivity. We have verified this hypothesis in two ways: by a sample of 500 already annotated NEs chosen randomly, and by another sample of 500 already annotated NEs from the two most semantically-close classes (location and organization). Both samples were revised by three other, independent native annotators and the percentages of corrections have been only 1.8% and 1.9%, respectively.

All NEs have been annotated in IOB format. The annotated dataset, **ArmanPersoNERCorpus**, contains 250,015 tokens and 7,682 sentences (considering the full-stop as the sentence terminator). It can be used to train NER systems in future research on Persian NER, but it also offers an ideal test set for evaluation of NER systems trained on silver standards. The NEs are categorized into six classes: *person*, *organization* (such as banks, ministries, embassies, teams, nationalities, networks and publishers), *location* (such as cities, villages, rivers, seas, golfs, deserts and mountains), *facility* (such as schools, universities, research centers, airports, railways, bridges, roads, harbors, stations, hospitals, parks, zoos and cinemas), *product* (such as books, newspapers, TV shows, movies, airplanes, ships, cars, theories, laws, agreements and religion), and *event* (such as wars, earthquakes, national holidays, festivals and conferences); *other* are the remaining tokens. It is worth noting that annotation was not trivial since individual tokens have been categorized according to the context. For instance, *"Tokyo"* is a different type of entity in sentence *"Tokyo$_{loc}$ is a beautiful city"* versus sentence *"London$_{org}$ and Tokyo$_{org}$ sign flight agreement"*. Table 1 summarizes the number of tokens for each entity class in ArmanPersoNERCorpus.

Figure 2 shows a snapshot of the dataset together with an English transliteration of the tokens. Each line contains five tab-separated columns. In order from left to right, they are ez$\bar{a}$fe, POS-tag, inflexion, token and NER-tag. The first three columns are inherited from the BijanKhan corpus. Ez$\bar{a}$fe [7] is a grammatical particle in the Persian language that connects words of a phrase, usually noun-phrase, together. It is pronounced as an unstressed i vowel between the linked words, but generally not indicated in writing.

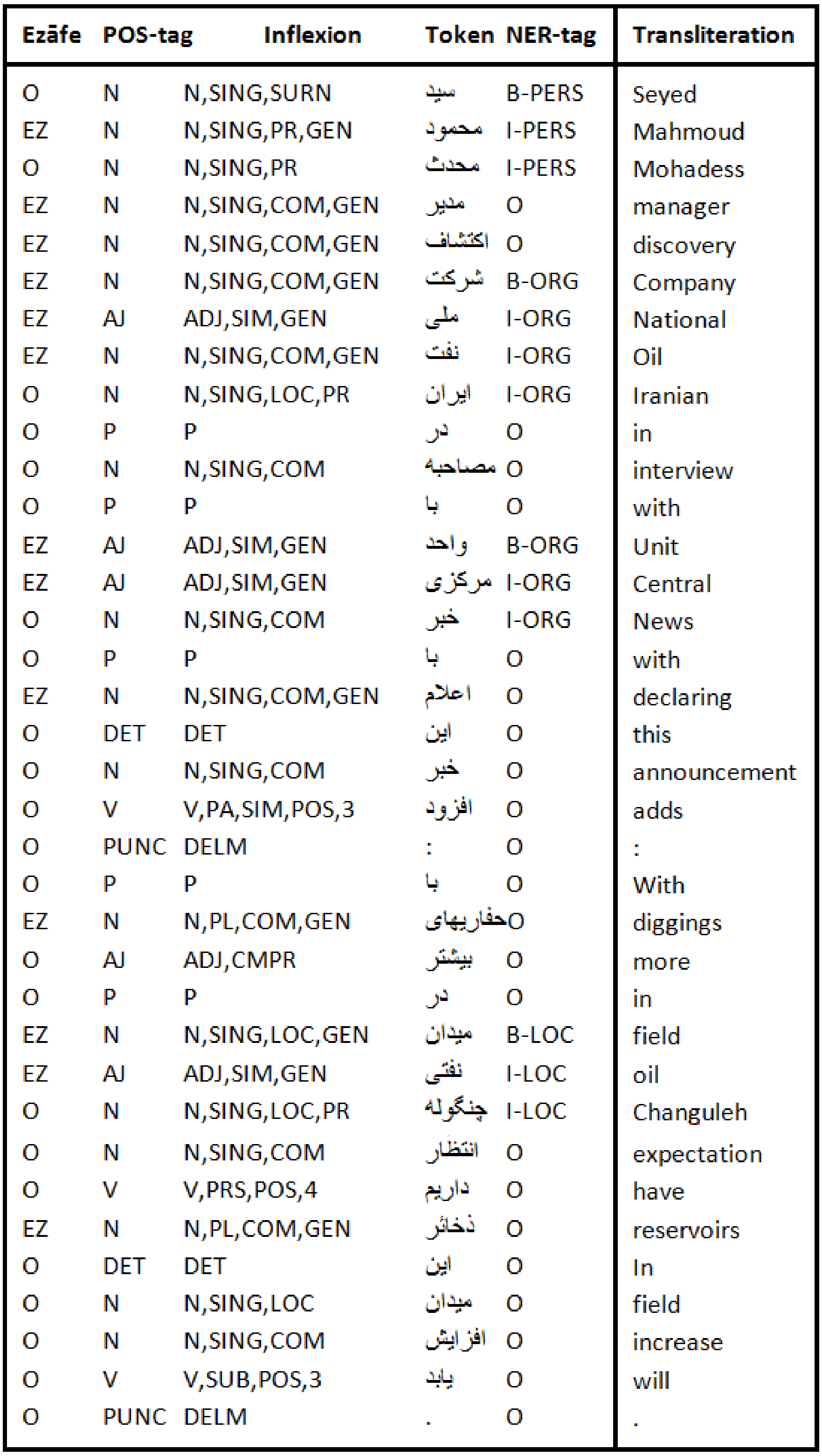
4.3 Text Normalization

As the preprocessing phase, the PersoSentencesCorpus has been normalized and tokenized following the approach proposed in (Feely et al., 2014) that suggests applying a pipeline of useful tools to deal with written Persian. The pipeline starts with PrePer (Seraji, 2013) which maps Arabic specific characters to their Persian Unicode equivalent. In addition, it replaces the full space between a word and its affix with a zero-width-non-joiner character. Then, a Farsi text normalizer (Feely, 2013) omits Arabic and Persian diacritics and unifies variant forms of some Persian characters to a single Unicode representation. Finally,

[5] http://armansoft.ir

[6] http://ece.ut.ac.ir/dbrg/bijankhan/

[7] https://en.wikipedia.org/wiki/Ezafe



Ezāfe	POS-tag	Inflexion	Token	NER-tag	Transliteration
O	N	N,SING,SURN	سید	B-PERS	Seyed
EZ	N	N,SING,PR,GEN	محمود	I-PERS	Mahmoud
O	N	N,SING,PR	محدث	I-PERS	Mohadess
EZ	N	N,SING,COM,GEN	مدیر	O	manager
EZ	N	N,SING,COM,GEN	اکتشاف	O	discovery
EZ	N	N,SING,COM,GEN	شرکت	B-ORG	Company
EZ	AJ	ADJ,SIM,GEN	ملی	I-ORG	National
EZ	N	N,SING,COM,GEN	نفت	I-ORG	Oil
O	N	N,SING,LOC,PR	ایران	I-ORG	Iranian
O	P	P	در	O	in
O	N	N,SING,COM	مصاحبه	O	interview
O	P	P	با	O	with
EZ	AJ	ADJ,SIM,GEN	واحد	B-ORG	Unit
EZ	AJ	ADJ,SIM,GEN	مرکزی	I-ORG	Central
O	N	N,SING,COM	خبر	I-ORG	News
O	P	P	با	O	with
EZ	N	N,SING,COM,GEN	اعلام	O	declaring
O	DET	DET	این	O	this
O	N	N,SING,COM	خبر	O	announcement
O	V	V,PA,SIM,POS,3	افزود	O	adds
O	PUNC	DELM	:	O	:
O	P	P	با	O	With
EZ	N	N,PL,COM,GEN	حفاریهای	O	diggings
O	AJ	ADJ,CMPR	بیشتر	O	more
O	P	P	در	O	in
EZ	N	N,SING,LOC,GEN	میدان	B-LOC	field
EZ	AJ	ADJ,SIM,GEN	نفتی	I-LOC	oil
O	N	N,SING,LOC,PR	چنگوله	I-LOC	Changuleh
O	N	N,SING,COM	انتظار	O	expectation
O	V	V,PRS,POS,4	داریم	O	have
EZ	N	N,PL,COM,GEN	ذخائر	O	reservoirs
O	DET	DET	این	O	In
O	N	N,SING,LOC	میدان	O	field
O	N	N,SING,COM	افزایش	O	increase
O	V	V,SUB,POS,3	یابد	O	will
O	PUNC	DELM	.	O	.

Figure 2: A snapshot of ArmanPersoNERCorpus.

	Entities													
	Person		**Organization**		**Location**		**Facility**		**Event**		**Product**		**Overall**	
Methods	MUC7	CoNLL	MUC7	CoNLL	MUC7	CoNLL	MUC7	CoNLL	MUC7	CoNLL	MUC7	CoNLL	MUC7	CoNLL
CRF	76.98	64.10	60.59	42.25	66.98	57.97	61.46	41.09	59.98	22.48	33.75	20.00	60.89	49.92
Jordan-RNN	79.13	72.13	67.31	57.28	69.90	62.70	63.49	51.92	62.30	39.79	49.50	**42.08**	68.53	60.52
SVM-HMM	**82.40**	**75.65**	**71.65**	**61.59**	**72.92**	**66.67**	**72.22**	**61.20**	**71.63**	**52.58**	**50.90**	41.37	**72.59**	**65.13**

Table 2: F_1 score comparison between three different classifiers based on MUC7 and CoNLL score functions for NER task on ArmanPersoNERCorpus. The F_1 score achieved by structural SVM is higher overall and for all classes but one, with the Jordan-RNN as the second best.

tokenization is performed by using three tokenizers in a cascade: the Farsi verb tokenizer of (Manshadi, 2013), SetPer (Seraji et al., 2012) and tok-tok (Dehdari, 2015).

5 Experiments

In this section, we report NER results based on the PersoSentencesCorpus and ArmanPersoNERCorpus datasets. The classification task is challenging given the much lower frequencies of the entity classes versus the non-entity class (*other*), as shown in Table 1. For this task, we have not used any of the additional linguistic information that is available from the dataset (such as POS tag, inflexion etc).

To calculate the co-occurrence matrix, C, we have used a context window of radius 5. The size of the dictionary, V, from the PersoSentencesCorpus is $|V| = 49,902$ and that of subset D is $D = 7,099$, obtained by selecting only the words with count greater than 15. The word embedding matrix $h(C)$ has been computed by heuristically setting $m = 300$. For classification, each word has been encoded as a 3-gram that includes the previous and following feature vectors. All the models used for classification share the same word embeddings.

For classification, we have compared the proposed SVM-HMM with a CRF and a deep learning approach based on the Jordan-RNN (Mesnil et al., 2013). For the SVM-HMM we have used structural SVM from (Joachims, 2008) with a Markov chain of order 3 and learning constant $C = 0.5$. The CRF is from the HCRF library (Morency et al., 2010) and is trained with an $L2$ regularizer of weight 100. The Jordan-RNN is a recurrent neural network from (Mesnil et al., 2013) trained with 100 hidden states and initialized using the same features vectors. All parameters were chosen by 3-fold cross-validation over a reasonable range of values. The indices for the three folds are available in the dataset to allow for future result comparison. We have also tried continuous bag of words (Mikolov et al., 2013), skip-grams (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) as embeddings, and the Elman-RNN (Mesnil et al., 2013) as classifier, but results have proved generally less accurate.

Table 2 shows the comparison of the average MUC7 and CoNLL scores from the 3-fold cross-validation for the three classifiers. The MUC7 and CoNLL scores are F_1 values adapted to the NER task, with the CoNLL score generally stricter than MUC7 (Nadeau and Sekine, 2007). As shown in Table 2, the scores achieved by the SVM-HMM are higher overall and for all classes but one, with the Jordan-RNN as the second best. To verify statistical significance, we have also run a paired t-test over the results from the six individual classes and confirmed statistical significance of the differences even at $p = 0.02$. The relative ranking between SVM-HMM and the CRF is supported by similar results in the literature, including (Nguyen and Guo, 2007; Tang et al., 2013; Lei et al., 2014), showing that regularized minimum-risk classifiers tend to outperform equivalent models trained under maximum conditional likelihood. The relative ranking between SVM-HMM and the RNN is instead somehow in contrast with the recent results in the literature, and a possible explanation for it is the relatively small size of the dataset compared to the number of free parameters in the models. We plan future comparative experiments with larger corpora to further probe this assumption.

6 Conclusion

In this paper, we have presented and released ArmanPersoNERCorpus, the first manually-annotated Persian NE dataset, and proposed an NER pipeline for the Persian language. The main components of the pipeline are word embedding by Hellinger PCA and classification by a structural SVM-HMM classifier. Experiments conducted over the ArmanPersoNERCorpus dataset have achieved interesting overall F_1 scores of 72.59 (MUC7) and 65.13 (CoNNL), higher than those of a CRF and a Jordan-RNN. The released dataset can be used for further development of Persian NER systems and for evaluation of systems trained on silver-standard corpora, and the achieved accuracy will provide a baseline for future comparisons.

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of 2015 SIAM International Conference on Data Mining*.

Maha Althobaiti, Udo Kruschwitz, and Massimo Poesio. 2015. Combining minimally-supervised methods for arabic named entity recognition. *TACL*, 3:243–255.

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

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

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

Weston Feely, Mehdi Manshadi, Robert E Frederking, and Lori S Levin. 2014. The cmu metal farsi nlp approach. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4052–4055, Reykjavik, Iceland.

Weston Feely. 2013. Open-source dependency parser, part-of-speech-tagger, and text normalizer for farsi (persian). `https://github.com/wfeely/farsiNLPTools/`.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 466–471, Kopenhagen.

Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam Yu. 2009. Predicting structured objects with support vector machines. *Communications of the ACM*, 52(11):97–104.

Thorsten Joachims. 2008. SVM^{hmm}: Sequence tagging with structural support vector machines. `https://www.cs.cornell.edu/people/tj/svm\_light/svm\_hmm.html/`.

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Remi Lebret and Ronan Collobert. 2014. Word embedding through hellinger pca. In *14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden.

Jianbo Lei, Buzhou Tang, Xueqin Lu, Kaihua Gao, Min Jiang, and Hua Xu. 2014. A comprehensive study of named entity recognition in chinese clinical text. *Journal of the American Medical Informatics Association*, 21:808–814.

Mehdi Manshadi. 2013. Farsi verb tokenizer. `https://github.com/mehdi-manshadi/Farsi-Verb-Tokenizer/`.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech 2013*, August.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, March.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Louis-Philippe Morency, C. Mario Christoudias, Ariadna Quattoni, Hugues Salamin, Giota Stratou, and Sybor Wang. 2010. Hidden-state conditional random field library. `http://multicomp.ict.usc.edu/?p=790`.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January. Publisher: John Benjamins Publishing Company.

Nam Nguyen and Yunsong Guo. 2007. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 681–688, New York, NY, USA. ACM.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artif. Intell.*, 194:151–175, January.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. `http://nlp.stanford.edu/projects/glove/`.

Lizhen Qu, Yi Zhang, Rui Wang, Lili Jiang, Rainer Gemulla, and Gerhard Weikum. 2014. Senti-lssvm: Sentiment-oriented multi-relation extraction with latent structural SVM. *TACL*, 2:155–168.

L.R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE Proc.*, 77:257–286.

Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. 2013. 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*, pages 306–314, Atlanta, Georgia, June. Association for Computational Linguistics.

Satoshi Sekine. 2007. The definition of sekines extended named entities. `http://nlp.cs.nyu.edu/ene/version7_1_0Beng.html`. New York University.

Mojgan Seraji, Megyesi Beta, and Nivre Joakim. 2012. A basic language resource kit for persian. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2245–2252, Istanbul, Turkey.

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

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

Buzhou Tang, Hongxin Cao, Yonghui Wu, Min Jiang, and Hua Xu. 2013. Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features. *BMC Medical Informatics and Decision Making*, 13(SUPPL1).

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. 2005. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 473–480, Stroudsburg, PA, USA. Association for Computational Linguistics.