Supervised Morphological Segmentation Using Rich Annotated Lexicon

Ebrahim Ansari,[†‡] **Zdeněk Žabokrtský,**[†]
Mohammad Mahmoudi,[‡] **Hamid Haghdoost**[‡] **and Jonáš Vidra**[†]
[†] Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics, Charles University
[‡] Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences
{ansari,m.mahmodi,hamid.h}@iasbs.ac.ir
{zabokrtsky,vidra}@ufal.mff.cuni.cz

Abstract

Morphological segmentation of words is the process of dividing a word into smaller units called morphemes; it is tricky especially when a morphologically rich or polysynthetic language is under question. In this work, we designed and evaluated several Recurrent Neural Network (RNN) based models as well as various other machine learning based approaches for the morphological segmentation task. We trained our models using annotated segmentation lexicons. To evaluate the effect of the training data size on our models, we decided to create a large hand-annotated morphologically segmented corpus of Persian words, which is, to the best of our knowledge, the first and the only segmentation lexicon for the Persian language. In the experimental phase, using the hand-annotated Persian lexicon and two smaller similar lexicons for Czech and Finnish languages, we evaluated the effect of the training data size, different hyper-parameters settings as well as different RNN-based models.

1 Introduction

Morphological analysis must be tackled somehow in all natural language processing tasks, such as machine translation, speech recognition, and information retrieval. Morphological segmentation of words is the process of dividing a word into smaller units called morphemes. Morphological segmentation task is harder for those languages which are morphologically rich and complex like Persian, Arabic, Czech, Finnish or Turkish, especially when there are not enough annotated data for those languages. In this paper, we designed and evaluated various supervised setups to perform morphological segmentation using a hand-annotated segmented lexicon for training.

The efficiency of supervised approaches (especially of deep neural network models) is naturally highly dependent on the size of training data. In order to evaluate the effect of the training data size on our segmentation models, we created a rich Persian hand-annotated segmentation lexicon, which is, as far as we know, the first and the only such computer-readable dataset for Persian. Persian (Farsi) is one of the languages of the Indo-European language family within the Indo-Iranian branch and is spoken in Iran, Afghanistan, Tajikistan and some other regions related to ancient Persian. In addition, we evaluated our models on Czech and Finnish, however, the amount of annotated data for them is substantially lower.

Automatic morphological segmentation was firstly introduced by Harris (1970). More recent research on morphological segmentation has been usually focused on unsupervised learning (Goldsmith, 2001; Creutz and Lagus, 2002; Poon et al., 2009; Narasimhan et al., 2015; Cao and Rei, 2016), whose goal is to find the segmentation boundaries using an unlabeled set of word forms (or possibly a corpus too). Probably the most popular unsupervised systems are LINGUISTICA (Goldsmith, 2001) and MORFESSOR, with a number of variants (Creutz and Lagus, 2002; Creutz et al., 2007; Grönroos et al., 2014). Another version of the latter which includes a semi-supervised extension was introduced by (Kohonen et al., 2010). Poon et al. (2009) presented a log-linear model which uses overlapping features for unsupervised morphological segmentation.

In spite of the dominance of the unsupervised systems, as soon as even just a small amount of

Proceedings of Recent Advances in Natural Language Processing, pages 52–61,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_007

segmented training data is available, then the entirely unsupervised systems tend not to be competitive. Furthermore, unsupervised segmentation still has considerable weaknesses, including over-segmentation of roots and erroneous segmentation of affixes (Wang et al., 2016). To deal with those limitations, recent works show a growing interest in semi-supervised and supervised approaches (Kohonen et al., 2010; Ruokolainen et al., 2013, 2014; Sirts and Goldwater, 2013; Wang et al., 2016; Kann and Schütze, 2016; Kann et al., 2018; Cotterell and Schütze, 2017; Grönroos et al., 2019) which employ annotated morpheme boundaries in the training phase.

In our work we designed and evaluated various machine learning models and trained them using only the annotated lexicon in a supervised manner. Our models do not leverage the unannotated data nor context information and only use the primary hand-annotated segmentation lexicons.

Experimental results show that our Bi-LSTM model perform slightly better than other models in boundary prediction for our hand-segmented Persian lexicon, while KNN (K-Nearest Neighbors algorithm) performs better when the whole word accuracy is under question.

The paper is organized as follows: Section 2 addresses the related work on morphological segmentation. Section 3 describes the methodology and machine learning models used in this work. Section 4 introduces our hand-segmented Persian lexicon as well as related preprocessing phases. Section 5 presents the experiment results compared to some other baseline systems and finally Section 6 concludes the paper.

2 Related Work

Supervised morphological segmentation, i.e. using a lexicon (or a corpus) with annotated morpheme boundaries in the training phase, has attracted increasing attention in recent years. One of the most recent successful research directions on supervised morphological segmentation is the work of (Ruokolainen et al., 2013), whose authors employ CRF (Conditional Random Fields), a popular discriminative log-linear model to predict morpheme boundaries given their local sub-string contexts instead of learning a morpheme lexicon. (Ruokolainen et al., 2014) extended their work to semi-supervised learning version by exploiting some available unsupervised segmentation techniques into their CRF-based model via a feature set augmentation. (Ruokolainen et al., 2014)

Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have recently achieved great success in sequence learning tasks, including outstanding results on sequential tasks such as machine translation (Sutskever et al., 2014). Wang et al. (2016) proposed three types of window-based LSTM neural network models named Window LSTM, Multi-window LSTMs and Bidirectional Multi-Window LSTMs, in order to automatically learn sequence structures and predict morphological segmentations of words in a raw text. They used only word boundary information without any need for extra feature engineering in the training phase. The authors compared their models with selected supervised models as well as with an LSTM architecture (Wang et al., 2016), and similarly to the work of Ruokolainen et al. (2013), their architecture is based on the whole text and context information instead of using only the lexicon. Cotterell and Schütze (2017) increased the segmentation accuracy by employing semantic coherence information in their models. They used RNN (Recurrent Neural Network) to design a composition model. They also found that using RNN with dependency vector has the best results on vector approximation (Cotterell and Schütze, 2017).

Recently, using encoder-decoder models Bahdanau et al. (2014) (attention-based models) made some great successes in machine translation systems. Kann and Schütze (2016) used an encoder-decoder model which encodes the input as a sequence of morphological tags of source and targets and feeds the model by sequence of letters of a source form. They select the final answer using a majority voting amongst their five different ensembled RNN encoder-decoder models. Kann and Schütze (2016), proposed a seq2seq (sequence-to-sequence network) architecture for the word segmentation task. They used a bi-directional RNN to encode the input word (i.e. sequence of characters) and concatenated forward and backward hidden states yielded from two GRUs and pass the result vector to decoder part. The decoder is a single GRU which uses segmentation symbols for training. She introduced two multi-task training approaches as well as data augmentations to improve the quality of the presented model. She shows that neural seq2seq models perform on par with or better than other strong baselines for polysynthetic languages in a minimal-resource setting. Their suggested neural seq2seq models constitute the state of the art for morphological segmentation in high-resource settings and for (mostly) European languages (Kann et al., 2018).

The main studied language in our work is Persian, which belongs to morphologically rich languages and which is powerful and versatile in word building. Having many affixes to form new words (over a hundred), and the ability to build affixes and especially prefixes from nouns, the Persian language is considered as an agglutinative language since it also frequently uses derivational agglutination to form new words from nouns, adjectives, and verbal stems. Hesabi (1988) claimed that Persian can derive more than 226 million words (Hesabi, 1988).

To the best of our knowledge, the research on morphology of the Persian language is very limited. Rasooli et al. (2013) claimed that performing morphological segmentation in the preprocessing phase of statistical machine translation could improve the quality of translations for morphology rich and complex languages. Although they segmented very low portion of Persian words (only some Persian verbs), the quality of their machine translation system increases by 1.9 points of BLEU score. Arabsorkhi and Shamsfard (2006) proposed a Minimum Description Length (MDL) based algorithm with some improvements for discovering the morphemes of Persian language through automatic analysis of corpora.

3 Our Machine Learning Models

In this work we decided to evaluate selected machine learning models including those feature-based machine learning approaches in which the task of word segmentation is reformulated as a classification task, as well as various deep-learning (DL for short) neural network models.

Because of huge number of learned parameters in DL, having enough training data is critical. The fact that we decided to create a large hand-annotated dataset for Persian allows evaluating the effect of the training data size on a relatively wide scale, as described in Section 4.

We convert all segmentations into a simple string format in which letters "B" and "L" encode the presence of the boundary letter and the continuation letter, respectively. For example for word "goes", the encoded segmentation is "LLBL", which shows that there is a segmentation boundary in front of the third letter ("e"). While in our model we consider only morphologically segmented lexicon and we do not employ any other information like corpus contexts or lists of unannotated words, this encoding is sufficient and make the specification of boundary location easy.

In the case of presence of a semi-space letter (a feature specific for the Persian written language), the semi-space letter is always considered as a boundary letter. An experiments focused on this feature is described in Subsection 5.2.3, which shows that our models could perform better when this information exists in the annotated lexicon.

3.1 Classification-Based Segmentation Models

In the first setup, we convert the segmentation task (the task of segmenting a word into a sequence morphemes) simply to a set of independent binary decisions capturing the presence or absence of a segmentation boundary in front of each letter in the word. For this task, we use various standard off-the-shelf classifiers available in the Scikit-learn toolkit (Pedregosa et al., 2011).

So far, we provide the classifiers only with features that are extractable from the word alone. More specifically, we use only character-based features. These character-based features include letters and letter sequences (and their combinations) before and after under the character under question, which is subsequently assigned one out of two classes: "B" for boundary characters, and "L" which stands for continuation characters. The main task of these methods is then to train a classification model to classify all characters in the word into those two classes, given binary features based on surrounding characters. For example, for the fifth character of word "hopeless", some of our features could be: "e", "le", and "ope". The classification predictions are performed independently.

3.2 Deep Neural Network Based Models

Besides the classification-based segmentation models, we designed and evaluated five DL models based on GRU, LSTM, Bi-LSTM, seq2seq and Bi-LSTM with the attention mechanism, respectively. The first three models are illustrated in Figures 1 and 2. The presented seq2seq model, is similar to the model described in (Grönroos et al., 2019). The last presented model is an attention based model, which is shown in Figure 3. In this model, we use Bi-LSTM as encoder and LSTM as attention layer, and finally, outputs of encoder and attention layers are added together.

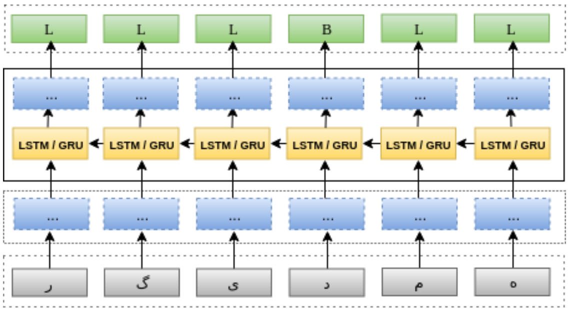

Figure 1: The schema of the LSTM/GRU models used in this experiments.

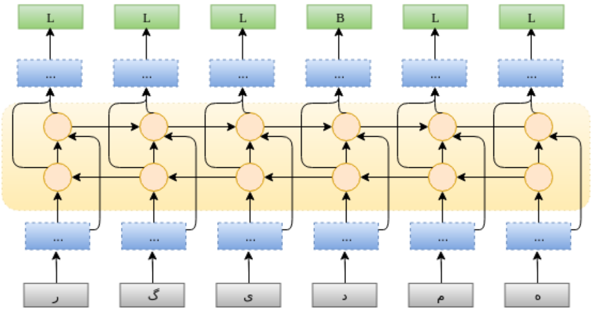

Figure 2: The schema of the Bi-LSTM model used in this experiments.

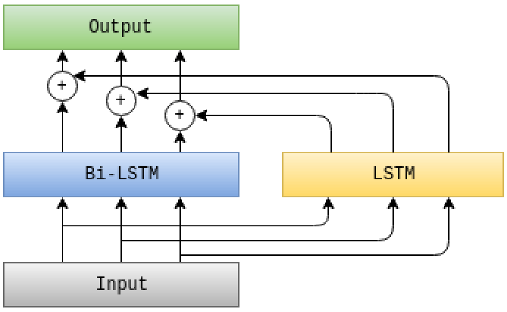

Figure 3: The schema of the Bi-LSTM with the attention mechanism model used in this experiments.

4 Morphological Segmentation Lexicons

In this section, the rich Persian hand-annotated dataset and the existing Finnish datasets from the Morpho-Challenge shared task 2010 (Virpioja et al., 2011) as well as the Czech dataset used in our experiments are described.

4.1 Persian Hand-Annotated Morphological Segmentation Dataset

We extracted our primary word list from three different corpora. The first corpus contains sentences extracted from the Persian Wikipedia (Karimi et al., 2018). The second one is popular Persian mono-lingual corpus BijanKhan (Bijankhan et al., 2011), and the last one is Persian-NER[1] (Poostchi et al., 2018).

For all introduced corpora, using Hazm toolset (Persian preprocessing and tokenization tools)[2] and the stemming tool presented by Taghi-Zadeh et al. (2015), we extracted and normalized all sentences and in the final steps using our rule-based stemmer and a Persian lemma collection, all words are lemmatized and stemmed. Finally all semi-spaces are automatically detected and fixed. Words with more than 10 occurrences in the corpora were selected for manual annotation. We decided to send all 80K words to our 16 annotators in the way that each word is checked and annotated by two independent persons. Annotators decided about the lemma of a word under question, segmentation parts, plurality, ambiguity (whether a word has more than one meaning) or they might delete the word if they think is not a proper Persian word. Moreover, some segmentations predicted by our automatic segmentator with high confidence score were offered to our annotators. We removed almost 30K words which were selected to be deleted by both annotators. And remaining 50K words sent for inter-annotation comparison part. In this step, all disagreements were checked and corrected by the authors of this paper and finally all words were quickly reviewed by two Persian linguists. The whole process took around six weeks. In order to use a hand-annotated lexicon in our work, we extracted the segmentation part from the dataset and converted it to our binary model which is described in Section 3.

The total number of words we used in our Persian dataset is 40K. The dataset is publicly available in the LINDAT/CLARIN repository (Ansari et al., 2019).

[1] https://github.com/HaniehP/PersianNER

[2] https://github.com/sobhe/hazm

4.2 Existing Finnish and Czech Segmentation Datasets

We downloaded the Finnish segmentation dataset from the Morpho-Challenge shared task 2010[3] (Virpioja et al., 2011) and converted them into our binary format. The Finnish dataset contains 2000 segmented words. While comparing to our hand-annotated Persian dataset these datasets are small, we used them to see the efficiency of our presented models when the size of training dataset is limited.

The Czech dataset results from a prototype segmentation annotation of Czech words. A sample of 1000 lemmas were selected randomly from DeriNet, which is a lexical database focus on derivation in Czech (Žabokrtský et al., 2016). The lemmas were manually segmented by two independent annotators, and all annotation differences were resolved subsequently during a third pass through the data. The annotation resulted in 4.6 morphemes per word, partially as a result of the fact that the lemmas were sampled uniformly, regardless of their corpus frequency, and thus the selection is biased towards longer words.

5 Experimental Results

To partition our dataset (Persian, Czech and Finnish) into training, development and test sets a commonly used method is used (Ruokolainen et al., 2013), which involves sorting words according to their frequency and assigning every eighth term starting from the first one into the test set and every eighth term from the second into the development set, while moving the remaining terms into the training set.

In order to evaluate the effect of the training data size, we randomly select the first 1/64, 1/32, 1/16, 1/8, 3/8, 1/4, 1/2, 3/4 and all amount of data from the training set to carry out experiments with different training sizes. In all experiments, we report three evaluation measures: the number of correctly predicted morpheme boundaries (in terms of precision, recall, and f-measure), the percentage of correct binary predictions on all characters, and the percentage of correctly segmented words.

As described in Section 2, some previous works reported accuracy in terms of the number of correct predictions (boundary and word) in a running text, instead of considering unique words sampled from a lexicon. Hence we decided to also report such accuracy in our experiments in addition to our lexicon evaluation. For this new experiment, we selected a part of a mono-lingual text and after removing all presented words in the text from our training lexicon, the remaining segmented words are considered as the training set and finally accuracy of word segmentation of words in test sentences is reported separately.

5.1 Baselines

We used two baseline systems which we selected to compare our models with. The first baseline is an unsupervised version of MORFESSOR, which is introduced and implemented by Creutz et al. (2007). The second baseline is FlatCat (Grönroos et al., 2014), which is a well-known semi-supervised version of MORFESSOR that uses the Hidden Markov Model for segmentation. In addition to the annotated data, semi-supervised MORFESSOR (i.e. FlatCat) uses a set of 100,000 word types following their frequency in the corpus as their unannotated training dataset. For both baselines, the best performing model is selected and compared with our neural network based models.

5.2 Results and Discussion

As described in Section 4, we designed various models for the morphological segmentation task. In the following subsections, different experiments done in this work are reviewed. In all tables, the column entitled by **W%** indicates the proportion of perfectly segmented words. The column entitled by **Ch%** indicated the accuracy of characters which are classified as boundary or non-boundary. Finally, **P%**, **R%**, and **F%** indicate precision, recall and F-measure score respectively for the morpheme boundary detection, naturally excluding the trivial final position characters from our evaluation.

5.2.1 Comparison of Different Models

Table 1 shows the evaluation results of morphological segmentation using our Persian hand-annotated dataset if the whole training data is used. For each model, only results of the best-performing hyperparameter configuration are reported. As is shown in Table 1, our Bi-LSTM model performs slightly better than the rest in boundary prediction, however, the classification models are surprisingly almost on the par with our complex DL model. Considering word accuracy,

[3] http://morpho.aalto.fi/events/morphochallenge2010/datasets.shtml

Model	**P% / R% / F%**	**W%**	**Ch%**
LSTM	90.09 / 87.55 / 88.80	64.10	93.20
GRU	85.43 / 84.50 / 84.96	58.35	91.44
Bi-LSTM	92.50 / 88.65 / **90.53**	66.51	94.37
Seq2Seq	88.04 / 84.04 / 86.09	59.10	91.65
Bi-LSTM with Attention	92.57 / 85.85 / 89.08	65.30	93.52
SVC, Kernel: linear	85.86 / 82.20 / 83.94	73.08	94.45
SVC, Kernel: poly, Degree: 2	89.57 / 85.86 / 87.67	**78.72**	95.72
SVC, Kernel: rbf	89.71 / 84.42 / 86.99	77.61	95.52
SVC, Kernel: poly, Degree: 5	89.77 / 83.91 / 86.74	77.17	95.45
SVC, Kernel: poly, Degree: 3	89.58 / 85.89 / 87.70	78.70	95.73
Logistic Regression, Solver: sag	87.55 / 79.66 / 83.42	72.60	94.39
Logistic Regression, Solver: liblinear	87.55 / 79.60 / 83.42	72.63	94.39
Logistic Regression, Solver: lbfgs	87.49 / 79.78 / 83.46	72.64	94.39
KNeighbors, Neighbors: 5	86.22 / 82.47 / 84.30	73.12	94.56
KNeighbors, Neighbors: 10	86.22 / 82.47 / 84.03	73.12	94.56
KNeighbors, Neighbors: 30	90.23 / 86.69 / 88.42	78.64	**95.73**
Ada Boost, Estimators: 100	83.34 / 64.10 / 72.46	58.21	90.83
Decision Tree	88.25 / 87.05 / 87.65	76.83	95.38
Random Forest, Estimators: 10	89.75 / 84.87 / 87.15	76.08	95.30
Random Forest, Estimators: 100	89.93 / 85.92 / 87.88	77.37	95.54
Bernoulli Naive Bayes	78.38 / 88.31 / 83.05	66.71	93.21
Perceptron MaxIteration: 50	83.98 / 74.45 / 78.93	65.07	92.52
Unsupervised MORFESSOR	69.58 / 81.10 / 74.90	29.01	83.28
Supervised MORFESSOR	82.13 / 92.94 / 87.20	59.56	91.60

Table 1: Result of applying our models on small Persian segmented lexicon. **P%**, **R%**, and **F%** indicate precision, recall and F-measure score respectively. **W%** means the percentage of number of correct predicted words and **Ch%** indicated the the accuracy of characters which are classified in two boundary or non-boundary classes.

Model	**P% / R% / F%**	**W%**	**Ch%**
LSTM	99.67 / 29.08 / 44.98	03.58	81.57
GRU	99.99 / 28.01 / 45.01	03.59	81.60
Bi-LSTM	86.96 / 32.82 / 47.66	04.88	81.30
Bi-LSTM with Attention	81.50 / 44.18 / 57.30	05.53	78.26
SVC, Kernel: linear	78.39 / 76.83 / 77.31	38.11	91.16
SVC, Kernel: poly, Degree: 2	89.00 / 77.62 / 82.23	47.55	93.63
SVC, Kernel: rbf	90.06 / 74.83 / 81.74	45.92	93.34
SVC, Kernel: poly, Degree: 5	91.35 / 64.71 / 75.75	35.83	91.75
SVC, Kernel: poly, Degree: 3	89.70 / 76.56 /**82.61**	**46.57**	**93.58**
Logistic Regression, Solver: sag	82.43 / 69.37 / 75.34	31.92	90.95
Logistic Regression, Solver: liblinear	82.43 / 69.37 / 75.34	31.92	90.95
Logistic Regression, Solver: lbfgs	82.43 / 69.37 / 75.34	31.92	90.95
KNeighbors, Neighbors: 5	82.56 / 71.23 / 76.48	33.55	91.27
KNeighbors, Neighbors: 10	82.56 / 71.23 / 76.48	33.55	91.27
KNeighbors, Neighbors: 30	82.56 / 71.23 / 76.48	33.55	91.27
Ada Boost, Estimators: 100	76.45 / 38.48 / 51.19	16.28	85.38
Decision Tree	79.58 / 76.29 / 77.90	39.41	91.38
Random Forest, Estimators: 10	87.41 / 68.44 / 76.77	37.45	91.75
Random Forest, Estimators: 100	88.08 / 72.83 / 79.73	44.29	92.62
Bernoulli Naive Bayes	64.27 / 76.43 / 69.82	26.38	86.84
Perceptron MaxIteration: 50	73.22 / 75.36 / 74.27	31.92	89.60
Unsupervised MORFESSOR	25.85 / 89.87 / 40.15	00.32	30.53
Supervised MORFESSOR	70.48 / 79.67 / 74.79	31.49	87.68

Table 2: Result of applying our models on small Finnish segmented lexicon.

Model	**P% / R% / F%**	**W%**	**Ch%**
LSTM	69.64 / 36.44 / 47.82	04.19	69.77
GRU	74.72 / 27.23 / 39.92	00.59	63.86
Bi-LSTM	68.56 / 48.33 / 56.69	05.38	67.45
Bi-LSTM with Attention	66.62 / 71.16 / 68.81	08.98	72.16
SVC, Kernel: linear	84.28 / 70.84 / 76.98	20.95	83.88
SVC, Kernel: poly, Degree: 2	91.42 / 69.46 / 78.94	31.73	85.90
SVC, Kernel: rbf	91.39 / 67.40 / 77.59	30.53	85.19
SVC, Kernel: poly, Degree: 5	94.03 / 48.71 / 64.18	20.35	79.32
SVC, Kernel: poly, Degree: 3	90.95 / 60.37 / 72.57	25.14	82.64
Logistic Regression, Solver: sag	90.69 / 66.89 / 76.99	25.04	84.80
Logistic Regression, Solver: liblinear	90.69 / 66.89 / 76.99	25.04	84.80
Logistic Regression, Solver: lbfgs	90.69 / 66.89 / 76.99	25.04	84.80
KNeighbors, Neighbors: 5	82.18 / 79.93 / 81.04	28.74	85.77
KNeighbors, Neighbors: 10	87.50 / 76.15 / **81.24**	**29.34**	**86.62**
KNeighbors, Neighbors: 30	82.18 / 79.93 / 81.04	28.74	85.77
Ada Boost, Estimators: 100	88.85 / 57.46 / 69.79	16.16	81.08
Decision Tree	78.46 / 56.26 / 65.53	15.56	77.49
Random Forest, Estimators: 10	91.42 / 65.86 / 76.57	**29.34**	84.67
Random Forest, Estimators: 100	91.76 / 68.78 / 76.82	**29.34**	85.77
Bernoulli Naive Bayes	85.94 / 74.44 / 79.77	26.94	85.64
Perceptron MaxIteration: 50	80.45 / 72.04 / 76.01	19.16	82.71
Unsupervised MORFESSOR	44.28 / 99.33 / 61.25	00.59	44.61
Supervised MORFESSOR	67.12 / 77.43 / 71.91	05.95	73.33

Table 3: Result of applying our models on the Czech segmented lexicon.

Model	**Parameters**	**P% / R% / F%**	**W%**	**Ch%**
Bi-LSTM	Outstate: 25 Dropout: 0.2	89.44 / 82.80 / 86.00	59.44	91.73
Bi-LSTM	Outstate: 50 Dropout: 0.2	88.79 / 87.89 / 88.34	62.57	92.86
Bi-LSTM	Outstate: 70 Dropout: 0.2	91.39 / 88.85 / 90.10	64.51	93.70
Bi-LSTM	Outstate: 70 Dropout: 0.5	92.50 / 88.65 / **90.53**	**66.51**	**94.37**
LSTM	Outstate: 25 Dropout: 0.2	91.69 / 83.00 / 87.13	62.32	92.45
LSTM	Outstate: 50 Dropout: 0.2	93.09 / 82.29 / 87.36	60.82	92.67
LSTM	Outstate: 70 Dropout: 0.2	90.09 / 87.55 / 88.80	64.10	93.20
LSTM	Outstate: 70 Dropout: 0.5	87.86 / 88.59 / 88.22	62.19	92.72

Table 4: Effect of using different hyperparameters on LSTM and Bi-LSTM models, two best performing deep neural network models for Persian dataset

classification models are performing better than DL models. A possible explanation for this is that the classification models make use of n-gram features and handle the characteristics of the whole word more efficiently than sequence-based models. Moreover, regarding our experiments, the presented seq2seq model does not perform well. An explanation could be that while there is not any available context information, the used attention mechanism does not have any far parts to make a relation between them. Moreover, our Bi-LSTM with the attention mechanism does not perform better than normal Bi-LSTM either. Finally, Tables 2 and 3 show the results of this experiment on two other languages, Finnish and Czech, for which the sizes of training data are very limited comparing the Persian dataset. As we expected, with so small training data, the classification methods perform better than more complex DL strategies.

Table 4 shows a comparison of our DL models, when different LSTM output sizes and drop-out thresholds are tested. Only two best-performing models (LSTM and Bi-LSTM) are shown.

As is seen in the tables, the classification models perform well when compared to more complex DL models. One explanation for this evidence is the lack of any external information (other than a segmented lexicon) which limits the number of possible features from the training data. For example there is no information about some previous words, and consequently RNN-based models can not learn any information about distant previous characters in the training phase. Possibly, this also explains the inferior performance of our seq2seq model compared to the Bi-LSTM model implemented for this work.

Finally, Table 5 shows results of selected models when the segmentation is done on all words occurring in a corpus instead of a segmented lexicon. In this experiments we expected those words with more frequency has higher effect on results comparing with less frequent words.

5.2.2 Effect of Training Data Size

In order to evaluate the effect of the training data size on our DL models, different amount of training data are selected from and feed to our models. Figure 4 and Figure 5 demonstrate an experiment in which the baseline line is the results of unsupervised version of MORFESSOR for similar test dataset. Only four best performing feature-based models in addition to two DL-based models are selected to be shown here. As this figure shows, after having more than 10K training instances, increasing the training data further does not have a substantial effect any more.

5.2.3 Semi-Space Feature for Persian Words

An important feature of the Persian and Arabic languages is the existence of semi-space. For example word "کتاب‌ها" (books) is a combination of word "کتاب" and "ها", in which the former is Persian translation of word "book" and the latter is morpheme for a plural form. We can say these semi-space signs segment words into smaller morphemes. However, in formal writing and in all Persian normal corpora, this space is neglected frequently and it could make a lot of problems in Persian and Arabic morphological segmentation task. For example both forms for the previous example, "کتاب‌ها" and "کتابها" , are considered correct in Persian text and have the same meaning. To deal with this problem and in order to improve the quality of our segmentation dataset, we implemented a preprocessor to distinguish this kind of space in Persian words and consequently our hand-annotated dataset contains these semi-spaces correctly. While we wanted to test the effect of having this prior knowledge in the lexicon, we evaluated our models in two different forms. In the first case, we used our hand annotated dataset as is. In the second case, we removed all semi-spaces from the lexicon. Table 6 shows a comparison for deploying our models on these two different datasets and as could be seen in this table, having the accurate dataset which is created by our preprocessing strategy could improve results drastically.

Model	P% / R% / F%	W%	Ch%
LSTM	94.42 / 92.93 / 93.67	78.14	95.13
Bi-LSTM	95.97 / 93.69 / **94.89**	**78.37**	**95.79**
SVC, Kernel: poly, Degree: 3	93.88 / 92.11 / 92.99	89.85	97.02
KNeighbors, Neighbors: 30	94.50 / 92.77 / 93.63	89.91	96.93
Random Forest, Estimators: 100	94.32 / 91.99 / 93.10	88.64	96.66

Table 5: Experiment results when a model is used to predict boundaries of Persian words of a small corpus instead of lexicon words. Only five best performing models are shown.

6 Conclusion

The main task of this work is to evaluate different supervised models to find the best segmentation of a word when only a segmented lexicon without any extra information is available in the training phase. In recent years, recurrent neural networks (RNN) attracted a growing interest in morphological analysis, that is why we decided to design and evaluate various neural network based models (LSTM, Bi-LSTM, GRU, and attention based models) as well as some machine learning classification models including SVM, Random Forest, Logistic Regression and others for our morphological segmentation task. While a critical point in any DL model is the training data size, we decided to create a rich hand annotated Persian lexicon which is the only segmented corpus for Persian words. Using this lexicon we evaluated our presented models as well as the effect of training data size on results. Moreover, we evaluated and tested our models on some limited datasets for Czech and Finnish languages. Experimental results show our Bi-LSTM model performs slightly better in boundary prediction, however the results of classification-based approaches overcome the DL models in percentage of completely correctly segmented words.

Acknowledgments

The research was supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495, International Mobility of Researchers at Charles University, and by grant No. 19-14534S of the Grant Agency of the Czech Republic. It has been using language resources developed, stored, and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the

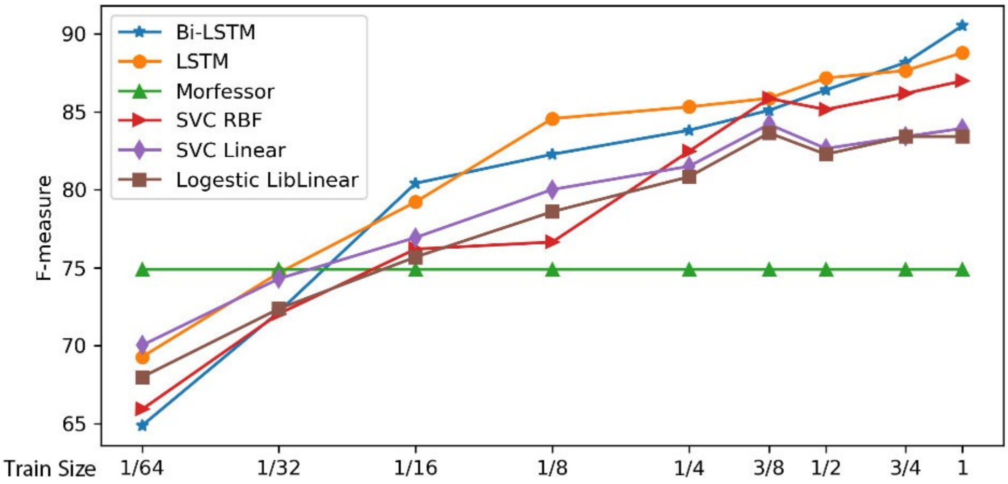

Figure 4: The effect of Persian training data size on boundary detection F-measure.

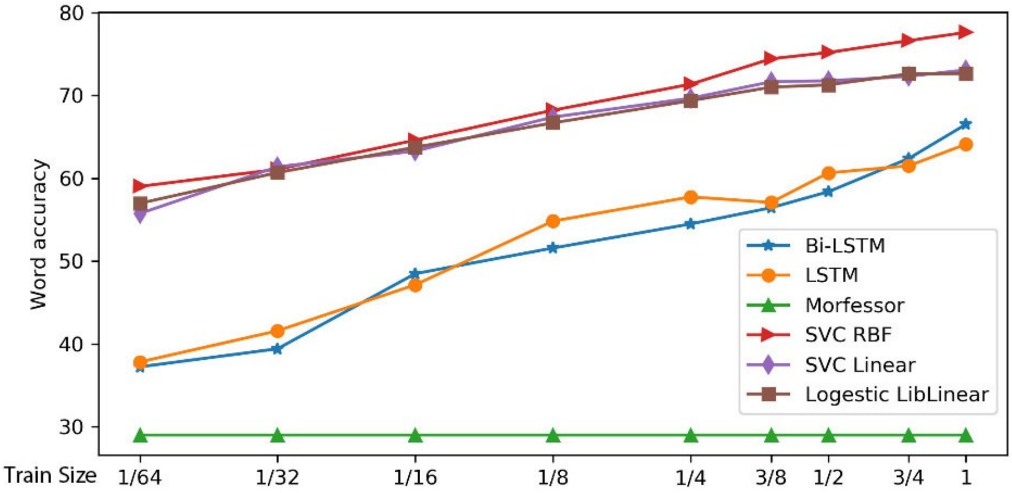

Figure 5: The effect of Persian training data size on whole-word segmentation accuracy.

Model	with semi			without semi		
	P% / R% / F%	W%	Ch%	P% / R% / F%	W%	Ch%
LSTM	90.09 / 87.57 / 88.80	64.10	93.20	91.15 / 74.76 / 82.15	51.42	89.53
Bi-LSTM	92.50 / 88.65 / **90.53**	66.51	94.37	89.19 / 77.18 / 82.75	52.58	89.64
SVC, Kernel: linear	85.86 / 82.20 / 83.94	73.08	94.45	81.67 / 77.75 / 79.66	68.17	92.62
SVC, Kernel: poly, Degree: 2	89.57 / 85.86 / 87.67	**78.72**	95.72	86.52 / 82.96 / 84.71	75.66	94.43
SVC, Kernel: rbf	89.71 / 84.42 / 86.99	77.61	95.52	86.34 / 80.96 / 83.56	74.39	94.08
SVC, Kernel: poly, Degree: 5	89.77 / 83.91 / 86.74	77.17	95.45	86.11 / 80.11 / 83.00	73.00	93.90
SVC, Kernel: poly, Degree: 3	89.58 / 85.89 / 87.70	78.70	**95.73**	86.30 / 83.02 / 84.63	75.30	94.39
Logistic Regression, Solver: sag	87.55 / 79.66 / 83.42	72.60	94.39	83.83 / 75.75 / 79.58	68.61	92.77
Logistic Regression, Solver: liblinear	87.55 / 79.60 / 83.42	72.63	94.39	83.84 / 75.75 / 79.59	68.63	92.78
Logistic Regression, Solver: lbfgs	87.49 / 79.78 / 83.46	72.64	94.39	83.74 / 75.59 / 79.46	68.47	92.73
KNeighbors, Neighbors: 5	82.47 / 86.22 / 84.30	73.12	94.56	82.19 / 76.34 / 79.15	67.36	92.52
KNeighbors, Neighbors: 10	86.22 / 82.47 / 84.30	73.12	94.56	82.19 / 76.34 / 79.15	67.36	95.52
KNeighbors, Neighbors: 30	90.23 / 86.69 / 88.42	78.64	95.73	82.19 / 76.34 / 79.15	67.36	92.52
Ada Boost, Estimators: 100	83.34 / 64.10 / 72.46	58.21	90.83	75.17 / 51.87 / 61.39	52.95	87.87
Decision Tree	88.25 / 87.05 / 87.65	76.83	95.38	88.24 / 86.05 / **87.13**	**75.92**	**95.21**
Random Forest, Estimators: 10	89.75 / 84.87 / 87.15	76.08	95.30	85.04 / 78.17 / 81.46	70.83	93.38
Random Forest, Estimators: 100	89.93 / 85.92 / 87.88	77.37	95.54	85.21 / 79.66 / 82.34	71.95	93.65
Bernoulli Naive Bayes	78.38 / 88.31 / 83.05	66.71	93.21	75.63 / 84.91 / 80.00	62.01	92.01
Perceptron MaxIteration: 50	83.98 / 74.45 / 78.93	65.07	92.52	75.41 / 77.28 / 76.34	62.51	90.05
Unsupervised MORFESSOR	69.58 / 81.10 / 74.90	29.01	83.28	71.16 / 81.88 / 76.14	30.33	83.48
Supervised MORFESSOR	82.13 / 92.94 / 87.20	59.56	91.60	81.60 / 92.24 / 86.60	58.84	90.80

Table 6: The effect of considering semi-space on training data when all training data are used.

Czech Republic (project LM2015071).

References

Ebrahim Ansari, Zdeněk Žabokrtský, Hamid Haghdoost, and Mahshid Nikravesh. 2019. Persian

Morphologically Segmented Lexicon 0.5. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, https://hdl.handle.net/11234/1-3011.

Mohsen Arabsorkhi and Mehrnoush Shamsfard. 2006. Unsupervised Discovery of Persian Morphemes. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters Demonstrations*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '06, pages 175–178. http://dl.acm.org/citation.cfm?id=1608974.1609002.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation. http://arxiv.org/abs/1409.0473.

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.

Kris Cao and Marek Rei. 2016. A joint model for word embedding and word morphology. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Berlin, Germany, pages 18–26. https://doi.org/10.18653/v1/W16-1603.

Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR* abs/1701.00946. http://arxiv.org/abs/1701.00946.

Mathias Creutz, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Trans. Speech Lang. Process.* 5(1):3:1–3:29. https://doi.org/10.1145/1322391.1322394.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*. Association for Computational Linguistics, pages 21–30. https://doi.org/10.3115/1118647.1118650.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Comput. Linguist.* 27(2):153–198. https://doi.org/10.1162/089120101750300490.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2019. North Sámi morphological segmentation with low-resource semi-supervised sequence labeling. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*. Association for Computational Linguistics, Tartu, Estonia, pages 15–26. https://www.aclweb.org/anthology/W19-0302.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 1177–1185. https://www.aclweb.org/anthology/C14-1111.

Zellig S. Harris. 1970. *From Phoneme to Morpheme*, Springer Netherlands, Dordrecht, pages 32–67. https://doi.org/10.1007/978-94-017-6059-1_2.

Mahmoud Hesabi. 1988. *Persian Affixes and Verbs*, volume 1. Javidan.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 47–57. https://doi.org/10.18653/v1/N18-1005.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics, Berlin, Germany, pages 62–70. https://doi.org/10.18653/v1/W16-2010.

Akbar Karimi, Ebrahim Ansari, and Bahram Sadeghi Bigham. 2018. Extracting an English-Persian Parallel Corpus from Comparable Corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*.

Oskar Kohonen, Sami Virpioja, Laura Leppänen, and Krista Lagus. 2010. Semi-supervised extensions to Morfessor Baseline.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics* 3:157–167. https://doi.org/10.1162/tacl_a_00130.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '09, pages 209–217. http://dl.acm.org/citation.cfm?id=1620754.1620785.

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, Miyazaki, Japan, May 7-12, 2018.*.

Mohammad Sadegh Rasooli, Ahmed El Kholy, and Nizar Habash. 2013. Orthographic and Morphological Processing for Persian-to-English Statistical Machine Translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, Nagoya, Japan, pages 1047–1051. https://www.aclweb.org/anthology/I13-1144.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Sofia, Bulgaria, pages 29–37. https://www.aclweb.org/anthology/W13-3504.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*. Association for Computational Linguistics, Gothenburg, Sweden, pages 84–89. https://doi.org/10.3115/v1/E14-4017.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *TACL* 1:255–266.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR* abs/1409.3215. http://arxiv.org/abs/1409.3215.

Hossein Taghi-Zadeh, Mohammad Hadi Sadreddini, Mohammad Hasan Diyanati, and Amir Hossein Rasekh. 2015. A new hybrid stemming method for Persian language. *Digital Scholarship in the Humanities* 32(1):209–221. https://doi.org/10.1093/llc/fqv053.

Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *TRAITEMENT AUTOMATIQUE DES LANGUES* 52(2):45–90.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological Segmentation with Window LSTM Neural Networks. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*. AAAI Press, pages 2842–2848.

Zdeněk Žabokrtský, Magda Ševčíková, Milan Straka, Jonáš Vidra, and Adéla Limburská. 2016. Merging data resources for inflectional and derivational morphology in Czech. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Paris, France, pages 1307–1314.