Building a Morphological Network for Persian on Top of a Morpheme-Segmented Lexicon

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

Abstract

In this work, we introduce a new large hand-annotated morpheme-segmentation lexicon of Persian words and present an algorithm that builds a morphological network using this segmented lexicon. The resulting network captures both derivational and inflectional relations. The algorithm for inducing the network approximates the distinction between root morphemes and affixes using the number of morpheme occurrences in the lexicon. We evaluate the quality (in the sense of linguistic correctness) of the resulting network empirically and compare it to the quality of a network generated in a setup based on manually distinguished non-root morphemes.

In the second phase of this work, we evaluated various strategies to add new words (unprocessed in the segmented lexicon) into an existing morphological network automatically. For this purpose, we created primary morphological networks based on two initial data: a manually segmented lexicon and an automatically segmented lexicon created by unsupervised MORFESSOR. Then new words are segmented using MORFESSOR and are added to the network. In our experiments, both supervised and unsupervised versions of MORFESSOR are evaluated and the results show that the procedure of network expansion could be performed automatically with reasonable accuracy.

1 Introduction

Even though the Natural Language community put more focus on inflectional morphology in the past, one can observe a growing interest in research on derivational morphology (and other aspects of word formation) recently, leading to existence of various morphological data resources. One relatively novel type of such resources are word-formation networks, some of which represent information about derivational morphology in the shape of a rooted tree. In such networks, the derivational relations are represented as directed edges between lexemes (Lango et al., 2018).

In our work, we present a procedure that builds a morphological network for the Persian language using a word segmentation lexicon. The resulting network (a directed graph) represents each cluster of morphologically related word forms as a tree-shaped component of the overall graph. The specific feature of such network is that it captures both derivational and inflectional relations. Figure 1 shows an example of such a tree for the Persian language which represents a base morpheme meaning “to know” and all derived and inflected descendants. In this example, the path from the root to one of the deepest leafs corresponds to the following meanings: (1) “to know", (2) “knowledge"/“science", (3) “scientist", (4) “scientists", (5) “some scientists".

What we use as a primary source of morphological information is a newly created manually annotated morpheme-segmented lexicon of Persian word forms, which is the only segmented lexicon for this language. At the same time, to the best of our knowledge, this lexicon could be considered as the biggest publicly available manually segmented lexicon at all (for any language).

Moreover, we expand the existing morphological network by adding new words into the current network by using our proposed core algorithm. In order to segment new words, we used both supervised and

Proceedings of the 2nd Int. Workshop on Resources and Tools for Derivational Morphology (DeriMo 2019), pages 91–100,
Prague, Czechia, 19-20 September 2019.

unsupervised version of MORFESSOR (Creutz et al., 2007; Grönroos et al., 2014), which is a popular automatic segmentation toolkit. After segmentation, the process of inducing morphological trees is the same as for hand-segmented words.

The paper is organized as follows: Section 2 addresses related work on derivational morphology networks and morphological segmentation. Section 3 introduces our hand-segmented Persian lexicon as well as related pre-processing phases. Section 4 describes the approach used in this work. Section 5 presents experiment results and finally Section 6 concludes the paper.

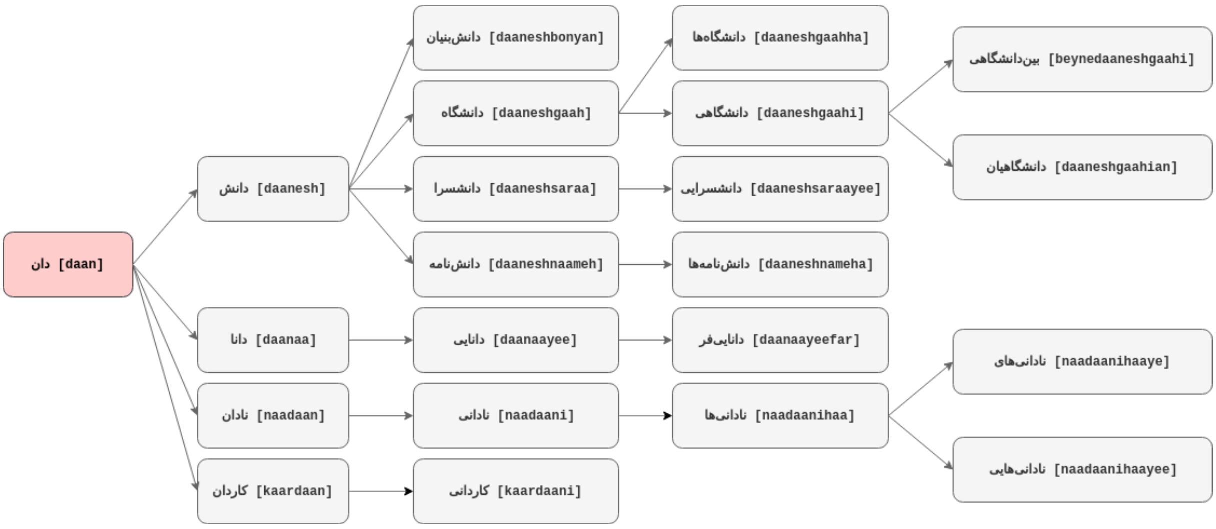

دان [daan]
دانش [daanesh]
دانا [daanaa]
نادان [naadaan]
کاردان [kaardaan]
دانش‌بنیان [daaneshbonyan]
دانشگاه [daaneshgaah]
دانشسرا [daaneshsaraa]
دانش‌نامه [daaneshnaameh]
دانایی [daanaayee]
نادانی [naadaani]
کاردانی [kaardaani]
دانشگاه‌ها [daaneshgaahha]
دانشگاهی [daaneshgaahi]
دانشسرایی [daaneshsaraayee]
دانش‌نامه‌ها [daaneshnameha]
دانایی‌فر [daanaayeefar]
نادانی‌ها [naadaanihaa]
بین‌دانشگاهی [beynedaaneshgaahi]
دانشگاهیان [daaneshgaahian]
نادانی‌های [naadaanihaaye]
نادانی‌هایی [naadaanihaayee]

Figure 1: A sample of a Persian morpohological tree for root [dan] which means "to know". The two children of the root node have the meaning of "knowledge" [danesh] and "smart" [dana], respectively. The path from the root to one of the deepest leaf corresponds to the following meanings: (1) "to know", (2) "knowledge"/"science", (3) "scientist", (4) "scientists", (5) "some scientists".

2 Related work

For some languages, intensive research exists with focus on construction of resources specialized in derivation, e.g. DerivBase (Zeller et al., 2013) for German, Démonette (Hathout and Namer, 2014) for French, DerivBase.Hr (Šnajder, 2014) for Croatian, DeriNet (Ševčíková and Žabokrtský, 2014; Žabokrtský et al., 2016) for Czech, (Vilares et al., 2001; Baranes and Sagot, 2014; Lango et al., 2018) for Spanish, Word Formation Latin (Litta et al., 2016), and (Piasecki et al., 2012; Kaleta, 2017; Lango et al., 2018) for Polish. However, for many other languages the data resources which provide information about derived words are scarce or lacking. Simultaneously, inflectional resources are further developed in recent years too (Hajič and Hlaváčová, 2013).

The language studied in our work is Persian, which belongs to morphologically rich languages and is powerful and versatile in word formation. Having many affixes to form new words (a few hundred), the Persian language is considered to be an agglutinative language since it also frequently uses derivational agglutination to form new words from nouns, adjectives, and verb stems. Hesabi (1988) claimed that Persian can derive more than 226 million word forms.

To our knowledge, research on Persian morphology is very limited. Rasooli et al. (2013) claimed that performing morphological segmentation in the pre-processing phase of statistical machine translation could improve the quality of translations for morphology rich and complex languages. Although they segmented only an extremely limited and non-representative sample of Persian words (tens of Persian verbs), the quality of their machine translation system increases by 1.9 points of BLEU score. Arabsorkhi and Shamsfard (2006) proposed an algorithm based on Minimum Description Length with certain improvements for discovering the morphemes of the Persian language through automatic analysis of corpora. However, since no Persian segmentation lexicon was made publicly available, we decided to create a manually segmented lexicon for Persian that contains 45K words now.

As we discussed before, we also trained and evaluated our methods using automatic morph-segmented data. Automatic morphological segmentation was firstly introduced by Harris (1955). 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.

3 Data: New Persian Segmented Lexicon

We extracted our primary word list from a collection composed of three corpora. The first corpus contains sentences extracted from the Persian Wikipedia (Karimi et al., 2018). The second one is a popular Persian corpus **BijanKhan** (Bijankhan et al., 2011), and the last one is the Persian Named Entity corpus[1] (Poostchi et al., 2018). For all those corpora, we used the **Hazm** toolkit (Persian pre-processing and tokenization tools)[2] and the stemming tool presented by Taghi-Zadeh et al. (2015). We extracted and normalized all sentences and lemmatized and stemmed all words using our rule-based stemmer and a lemmatizer that uses our collection of Persian lemmas. Finally all semi-spaces are automatically detected and fixed. 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.

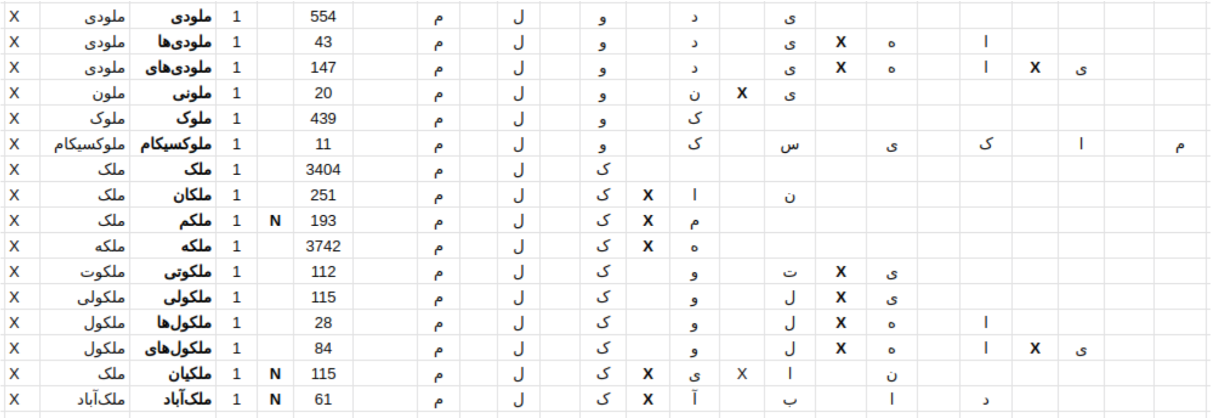
Words with more than 10 occurrences in our corpus collection were selected for manual annotation, which resulted in a set of around 90K word forms. We distributed them among 16 annotators in a way that each word was checked and annotated by two persons independently. Annotators made decisions about the lemma of a word under question, segmentation parts, plurality, and ambiguity (whether a word had more than one meaning). The manual annotation of segmentation was accelerated by predicting morpheme boundaries by our automatic segmenter and offering the most confident boundaries to the annotators. The annotators were allowed to indicate that a word was not a proper Persian word, while we decided to remove all borrowing words which are not common in the Persian language. For all disagreement in deleted words list, a third reviewer made the final decision for word removing. The whole process led to removing almost 30K words from the lexicon.

The remaining words were sent for resolving inter-annotator differences. All disagreements were reviewed and corrected by the authors of this paper. Finally all annotated words were quickly reviewed by two Persian linguists. The whole process took almost six weeks. Figure 2 shows a snapshot of our morpheme-segmented dataset.

In order to use a hand-annotated lexicon in our work, we extracted the segmentation part from the dataset and converted it into our binary model which is suitable for our algorithm described in Section 4. The total number of words we used in our Persian dataset was 45K. Finally, in order to make the data more appropriate for future segmentation experiments, we divided it into three different sets. The training set includes almost 37K, both test and development sets includes around 4K words each. Moreover, we divided the dataset based on their derivational trees which makes it possible to have all words with the same root in the same set. The dataset which is a rich test set for future experiments on Persian morphological tasks is publicly available in the LINDAT/CLARIN repository (Ansari et al., 2019).

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

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



X	ملودی	**ملودی**	1		554		م		ل		و		د		ی								
X	ملودی	**ملودی‌ها**	1		43		م		ل		و		د		ی	X	ه		ا				
X	ملودی	**ملودی‌های**	1		147		م		ل		و		د		ی	X	ه		ا	X	ی		
X	ملون	**ملونی**	1		20		م		ل		و		ن	X	ی								
X	ملوک	**ملوک**	1		439		م		ل		و		ک										
X	ملوکسیکام	**ملوکسیکام**	1		11		م		ل		و		ک		س		ی		ک		ا		م
X	ملک	**ملک**	1		3404		م		ل		ک												
X	ملک	**ملکان**	1		251		م		ل		ک	X	ا		ن								
X	ملک	**ملکم**	1	**N**	193		م		ل		ک	X	م										
X	ملکه	**ملکه**	1		3742		م		ل		ک	X	ه										
X	ملکوت	**ملکوتی**	1		112		م		ل		ک		و		ت	X	ی						
X	ملکولی	**ملکولی**	1		115		م		ل		ک		و		ل	X	ی						
X	ملکول	**ملکول‌ها**	1		28		م		ل		ک		و		ل	X	ه		ا				
X	ملکول	**ملکول‌های**	1		84		م		ل		ک		و		ل	X	ه		ا	X	ی		
X	ملک	**ملکیان**	1	**N**	115		م		ل		ک	X	ی	X	ا		ن						
X	ملک‌آباد	**ملک‌آباد**	1	**N**	61		م		ل		ک	X	آ		ب		ا		د				

Figure 2: A snapshot of the annotated dataset.

4 Morphological Network Construction

In this section, the method used in our work is described. Subsection 4.1 introduces our algorithm developed for the task and Subsection 4.2 describes the idea of using automatic segmented lexicon.

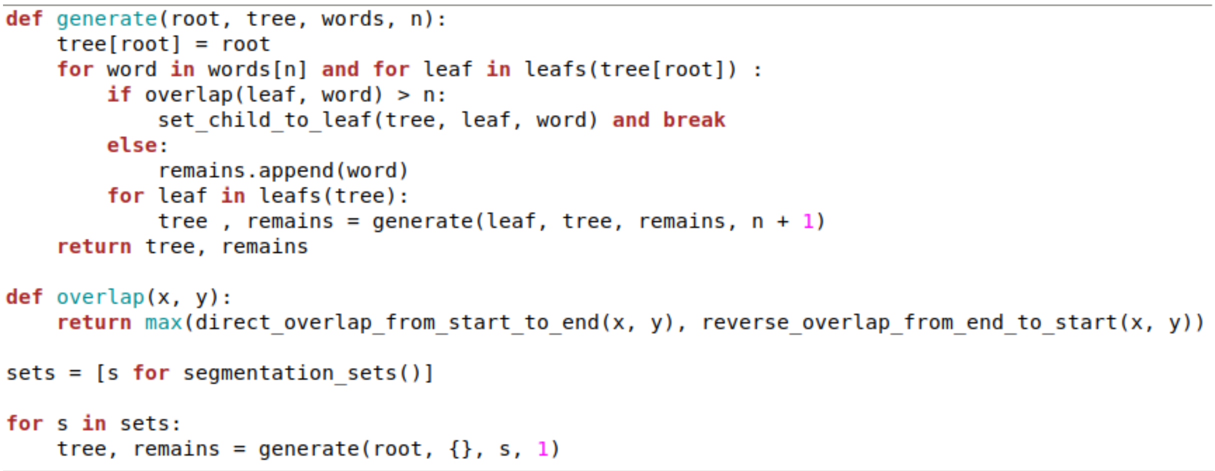
4.1 Automatic Network Construction

The core idea of this work is to construct a morphological network using a morpheme-segmented lexicon. First we need to partition the set of word forms into subsets based on same root morphemes. We approximate the distinction between root morphemes and affixes using the frequency of individual morphemes in the segmented lexicon. After calculating the frequencies, the m most frequent segments (we used 100 and 200 for m in our experiments) are removed from the set of potential root morphemes; all the remaining morphemes are stored in a set named $roots$. While the first m frequent segments are repeated more than other segments in our dataset, usually they are not root morphemes and could be considered as affixes. Table 1 shows an example of the most frequent segments based on our Persian segmented lexicon; none of them are none-root morphemes.

The next phase is to add nodes to our morphological graph (i.e., the network contains morphological trees) based on the assembled set of root morphemes. For each r_i from the $roots$ set, we create a set of words that contain r_i. We name this set $words_i$. Now, we add r_i as a new node to our derivational graph. In the next step, we find and connect all the words in $words_i$ in the network. We divide all the words in $words_i$ into n smaller sets $words_{i,2}, words_{i,3}, ..., words_{i,n}$ based on the number of their segments. The set $words_{i,j}$ includes all words containing r_i and their number of segments is equal to j. First, we check all w in $words_{i,2}$ and if it contains a node in the tree that includes r_i, we add it to the network graph, otherwise we add w to the $remaining$ set. Then, for the next group, $words_{i,3}$, we follow a similar procedure, however, we add all w in $words_{i,3}$ when it contains a node existing in $words_{i,2}$ (i.e., set of words with two segments). Then we add them to $remaining$ if there is not any subset in our current graph. We iterate this procedure until we pass all sets. Now, for each w in $remaining$ set, we check all added nodes and add w as a child of any node with maximum number of segments. It means it would be connected to the root if there is no other option available. Figure 3 shows a simple pseudo code of the segmentation graph generating procedure. the $generate$ function is recursive and gets $root$, current $tree$, remaining $words$ and current $step$ as the input parameters and returns a new $tree$ and remaining $words$. The $overlap$ function gets two words as the input and checks direct and reverse overlap count of the morphemes and returns maximum of them.

4.2 Semi-automatic Network Construction

After our primary experiments, we observed some root morphemes such as [shah] "king" (clearly not an affix) among the first 200 frequent segments. In order to quantify the influence of such wrongly classified affixes, we performed a modified versions of the above described experiment. This time, after frequency



```
def generate(root, tree, words, n):
    tree[root] = root
    for word in words[n] and for leaf in leafs(tree[root]) :
        if overlap(leaf, word) > n:
            set_child_to_leaf(tree, leaf, word) and break
        else:
            remains.append(word)
        for leaf in leafs(tree):
            tree , remains = generate(leaf, tree, remains, n + 1)
    return tree, remains

def overlap(x, y):
    return max(direct_overlap_from_start_to_end(x, y), reverse_overlap_from_end_to_start(x, y))

sets = [s for segmentation_sets()]

for s in sets:
    tree, remains = generate(root, {}, s, 1)
```

Figure 3: A pseudo-code of generating derivational graphs.

counting, we selected the m most frequent morphemes and two annotators decided in parallel whether they are root morphemes or not (such annotation is not a time-consuming task for a human at all). The rest of the experiment remained the same. Again, we set m equal to 100 or 200.

Table 1: 40 most frequent morphemes in the hand-segmented segmented lexicon.

rank	segment	freq.	rank	segment	freq.	rank	segment	freq.	rank	segment	freq.
1	ی [y]	9118	11	ای [ee]	583	21	هم [ham]	278	31	است [ast]	216
2	ها [haa]	4819	12	ال [al]	561	22	ید [id]	274	32	ش [ash]	206
3	ه [h]	2898	13	تر [tar]	746	23	ا [aa]	274	33	دان [daan]	198
4	ان [aan]	1708	14	ات [aat]	425	24	م [m]	267	34	شان [shaan]	193
5	می [mi]	1112	15	ب [b]	422	25	در [dar]	260	35	گاه [gaah]	192
6	یی [yee]	941	16	ین [een]	396	26	کار [kaar]	258	36	کن [kan]	189
7	ش [sh]	891	17	ده [deh]	383	27	ساز [saaz]	254	37	پر [por]	187
8	ن [n]	864	18	شد [shod]	359	28	دو [do]	241	38	نا [naa]	178
9	ند [nd]	782	19	دار [daar]	337	29	بر [bar]	239	39	ت [t]	173
10	د [d]	658	20	و [oo]	308	30	گر [gar]	232	40	شاه [shaah]	164

4.3 Automatic Network Expansion Using Morpheme-Segmented Data Created by MORFESSOR

In this part of our work, we decided to propose an automatic procedure to expand the existing derivational network by adding selected new and unseen words into the graph. In other words, when the primary network is ready, we try to add new words into it using the core algorithm explained in Section 4.1. However, the segmentation process for these new words is done by MORFESSOR. Figure 4 shows a flowchart of segmentation process workflow.

As is shown in Figure 4, the effect of using MORFESSOR could be evaluated in two different ways. First, in the initial data segmentation which is used to create the primary morphological network. Second way of adopting MORFESSOR is when we have some new words (i.e. test words) and we want to add them into our existing network and we can use MORFESSOR to segment them in an automatic way. In other words, in the testing phase, we have words that do not exist in our hand-annotated dataset and for creating derivational network of morphemes we need a segmentation for them. In order to resolve this problem, we decided to use an automatic segmentation algorithm to segment these unseen words and we selected MORFESSOR for this purpose. It works in two ways; supervised and unsupervised: we created two models of MORFESSOR and in the testing phase when a new word is under question, we segment it and add it to our existing tree based on that segmentation.

In this experiment, the unsupervised model is created based on all 97K raw data that we collected in our work and supervised MORFESSOR is trained using the 45K hand-annotated dataset. Experimental results in Section 5 show that the supervised model has better performance in comparison with the unsupervised one in the final tree accuracy.

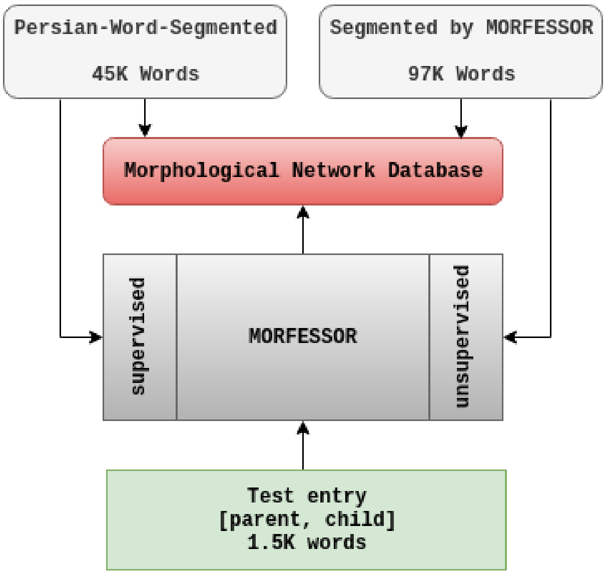


Figure 4: Morphological Network Database construction flowchart which shows the primary network construction and the expansion procedure.

5 Experiments

In order to estimate the quality of the resulting network, we randomly selected 400 nodes and checked if their parent node is identified correctly. We ran our automatic and semi-automatic versions of the algorithm using two thresholds for skipped root morphemes, 100 and 200. Table 2 summarizes the results for the individual experiment configurations. In all cases, the number of nodes in the generated graphs is 45K, which is equal to the total number of words in our manually segmented lexicon. Finally, Figure 5 shows three sample sub-graphs extracted by our algorithms.

Table 2: Accuracy for both automatic and semi-automatic methods using different numbers of non-roots in primary phase on 400 randomly selected nodes (i.e., words).

non-root selection	# of non-roots	accuracy
automatic	100	89.5%
automatic	200	86.3%
semi-automatic	100	91.0%
semi-automatic	200	92.8%

In the next experiment, we tried to evaluate our strategy to expand the morphological network when new unseen words were supposed to be added to the graph. Table 3 shows results of eight configurations of our experiments with using MORFESSOR as the automatic morpheme segmentation tool. In the first half of the table, we used all available words to create out initial network and to make the segmentation, the unsupervised version of MORFESSOR is used. In the bottom half of Table 3, all rows show the results when the hand-annotated segmented data is used. Similarly to the previous experiment, we removed and

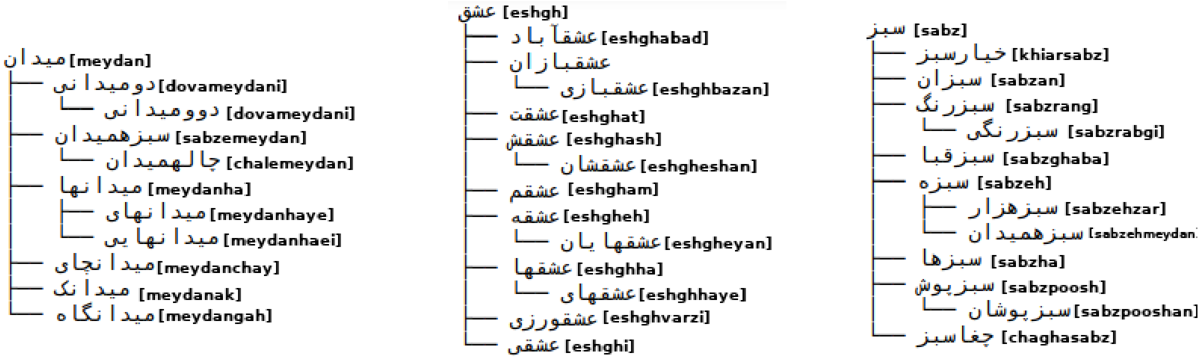


Figure 5: Samples of trees generated by our procedures describe in Sections 4.1 and 4.1.

cleaned most frequent non-root morphemes in two ways: in automatic removing during which we ignore all first 200 frequent morphemes, and in manual removing during which the selection and removing is done by an annotator. In other words, the first two columns of this table represents the configuration of the initial tree creation. The third column of Table 3 represents the method we used for segmenting the new words and in this column. Caption "Supervised" declares we used supervised MORFESSOR which is trained using 45K hand-annotated data and "Unsupervised" indicates that the segmentation is done by using fully unsupervised version of MORFESSOR. For all tests in this experiment, we provided a hand-annotated morphological network with 1500 words.

Table 3: Accuracy for tree structures on 1.2K dataset.

init. network creation	non-root selection	test words segmentation	Accuracy
97K/Segmented by MORFESSOR	automatic	sup. MORFESSOR	0.893
97K/Segmented by MORFESSOR	automatic	uns. MORFESSOR	0.777
97K/Segmented by MORFESSOR	manual	sup. MORFESSOR	0.893
97K/Segmented by MORFESSOR	manual	uns. MORFESSOR	0.777
45K Persian-Word-Segmented	automatic	sup. MORFESSOR	0.919
45K Persian-Word-Segmented	automatic	uns. MORFESSOR	0.846
45K Persian-Word-Segmented	manual	sup. MORFESSOR	0.934
45K Persian-Word-Segmented	manual	uns. MORFESSOR	0.866

5.1 Error Analysis

In this section we present an error analysis based on our observations. In the first experiment, when we created a morphological network using the hand-segmented lexicon and the whole procedure was automatic (Section 4.1), we explored two different error types. The first one happened when we wrongly labeled a root morpheme as the non-root which was ranked among top frequent morphemes. For example, as can be seen in in Table 1, the word "شاه [shaah]" which means "king" and ranked 40 is a root morpheme, but we automatically labeled it as a non-root. The second common type of errors happened when our method classified a non-root morpheme as a root morpheme. For example, morpheme "ون [oon] (plural suffix)" was classified wrongly as a root morpheme by our algorithm.

In the second experiment (Section 4.2), we solved the first problem by checking the frequent morphemes manually, and as we expected, the accuracy of the result was better comparing with automatic non-root selection. However, the second problem (false roots) still existed. The main reason of this problem is that there are not enough words in our segmented lexicon, and thus our algorithm is not able to identify correct parts of rare words as their root morphemes.

In our last experiment (i.e. expanding the existing graph by adding the new unseen words) which is described in Section 4.3 the main reason of seen errors was the wrong segmentation for some new test words. It means in some cases MORFESSOR did the segmentation wrong which consequently led to wrong morpheme detection and wrong parent/child identification. Table 4 shows five examples of wrong segmentation of supervised and unsupervised MORFESSOR for our test words. Moreover, in some cases, there was not any child and parent word for test words and consequently our algorithm could not expand the graph correctly based on them. However, this error happened very few times while our primary graph was big enough.

Table 4: Sample segmentation of supervised and unsupervised MORFESSOR for test words.

word	correct segmentation	unsup. MORFESSOR	sup. MORFESSOR
آبزی [aabzi]	آب-زی	آبزی	آب-ز-ی
آبشش‌ها [aabshoshha]	آب-شش-ها	آبشش-ها	آب-ش-ش-ها
تعهدنامه [taahodnameh]	تعهد-نامه	ت-عهدنامه	ت-عهد-نامه
بی‌اجازه [biejaazeh]	بی-اجازه	ب-ی-اجازه	ب-ی-اجازه
حاکمیت [haakemiat]	حاکم-یت	حاکمیت	ح-اک-میت

6 Conclusions and future work

In this work, we developed and empirically evaluated an algorithm for creating a morphological (derivational and inflectional) network using a morpheme-segmented lexicon. Our algorithm tries to find all root candidates automatically and creates connections for all words of the lexicon. In addition, we evaluated a modification of our procedure based on hand-validated set of non-root morphemes. To prepare input for our presented algorithm, we presented a large manually annotated Persian lexicon which is the only segmented corpus for Persian words and which currently includes 45K words. In the second part of this work, we tried to expand the morphological network by adding 1500 new words into the existing network. While this procedure is automatic, we tried to segment new test words using both supervised and unsupervised versions of MORFESSOR, the automatic segmentation toolkit. These segmented morphemes are used as the input of our proposed algorithm to find the parents of new 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/CLARIAH CZ project (LM2015071, LM2018101).

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 &#38; Demonstrations*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '06, pages 175–178. http://dl.acm.org/citation.cfm?id=1608974.1609002.

Marion Baranes and Benoît Sagot. 2014. A language-independent approach to extracting derivational relations from an inflectional lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 2793–2799.

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.

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. *Computational Linguistics* 27(2):153–198. https://doi.org/10.1162/089120101750300490.

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.

Jan Hajič and Jaroslava Hlaváčová. 2013. MorfFlex CZ. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. http://hdl.handle.net/11858/00-097C-0000-0015-A780-9.

Zellig Harris. 1955. From phoneme to morpheme. *Language* 31:209–221.

Nabil Hathout and Fiammetta Namer. 2014. Démonette, a French derivational morpho-semantic network. *Linguistic Issues in Language Technology* 11(5):125–168.

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

Zbigniew Kaleta. 2017. Automatic Pairing of Perfective and Imperfective Verbs in Polish. In *Proceedings of the 8th Language and Technology Conference*.

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. In *Proceedings of the Morpho Challenge 2010 Workshop*. pages 30–34.

Mateusz Lango, Magda Ševčíková, and Zdeněk Žabokrtský. 2018. Semi-Automatic Construction of Word-Formation Networks (for Polish and Spanish). In *Proceedings of the 11th Language Resources and Evaluation Conference*. European Language Resource Association, Miyazaki, Japan. https://www.aclweb.org/anthology/L18-1291.

Eleonora Litta, Marco Passarotti, and Chris Culy. 2016. Formatio formosa est. Building a Word Formation Lexicon for Latin. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*. http://ceur-ws.org/Vol-1749/paper32.pdf.

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.

Maciej Piasecki, Radoslaw Ramocki, and Marek Maziarz. 2012. Recognition of Polish Derivational Relations Based on Supervised Learning Scheme. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), Istanbul, Turkey, pages 916–922.

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.

Magda Ševčíková and Zdeněk Žabokrtský. 2014. Word-formation network for Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 1087–1093.

Jan Šnajder. 2014. DerivBase.hr: A high-coverage derivational morphology resource for Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 3371–3377.

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.

Jesús Vilares, David Cabrero, and Miguel A. Alonso. 2001. Applying Productive Derivational Morphology to Term Indexing of Spanish Texts. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 336–348.

Zdeněk Žabokrtský, Magda Ševčíková, Milan Straka, Jonáš Vidra, and Adéla Limburská. 2016. Merging data resources for inflectional and derivational morphology in Czech. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Portorož, Slovenia, pages 1307–1314. https://www.aclweb.org/anthology/L16-1208.

Britta Zeller, Jan Šnajder, and Sebastian Padó. 2013. DErivBase: Inducing and evaluating a derivational morphology resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1201–1211. https://www.aclweb.org/anthology/P13-1118.