EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

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

HAMID HAGHDOOST, *EBRAHIM ANSARI*, ZDENĚK ŽABOKRTSKÝ, MAHSHID NIKRAVESH

INSTITUTE FOR ADVANCED STUDIES IN BASIC SCIENCES (IASBS), IRAN
INSTITUTE OF FORMAL AND APPLIED LINGUISTICS (UFAL), CHARLES UNIVERSITY, CZECH REPUBLIC

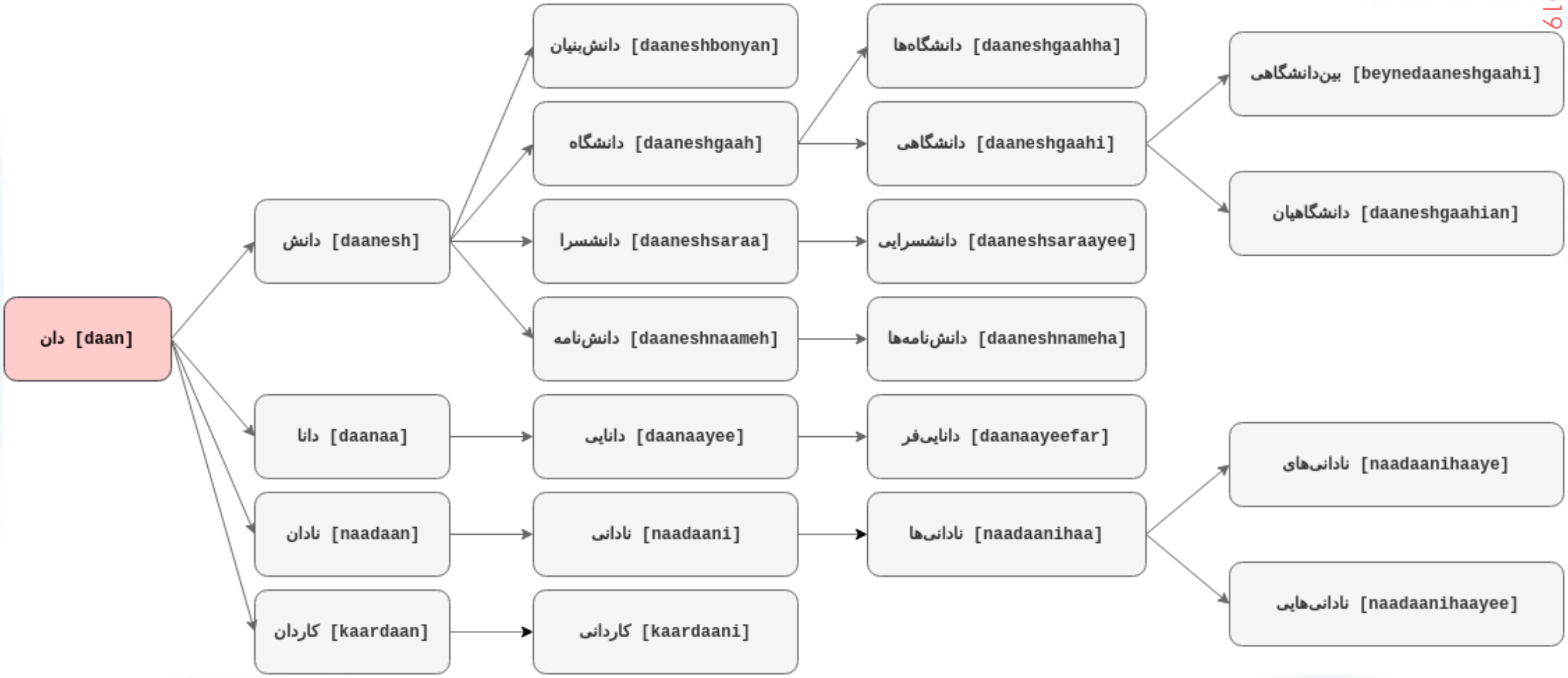
outline

- introduction
  - definitions
  - selected language: Persian
- data preparation
- morphological network construction
- morphological network expansion
- error analysis
- conclusion

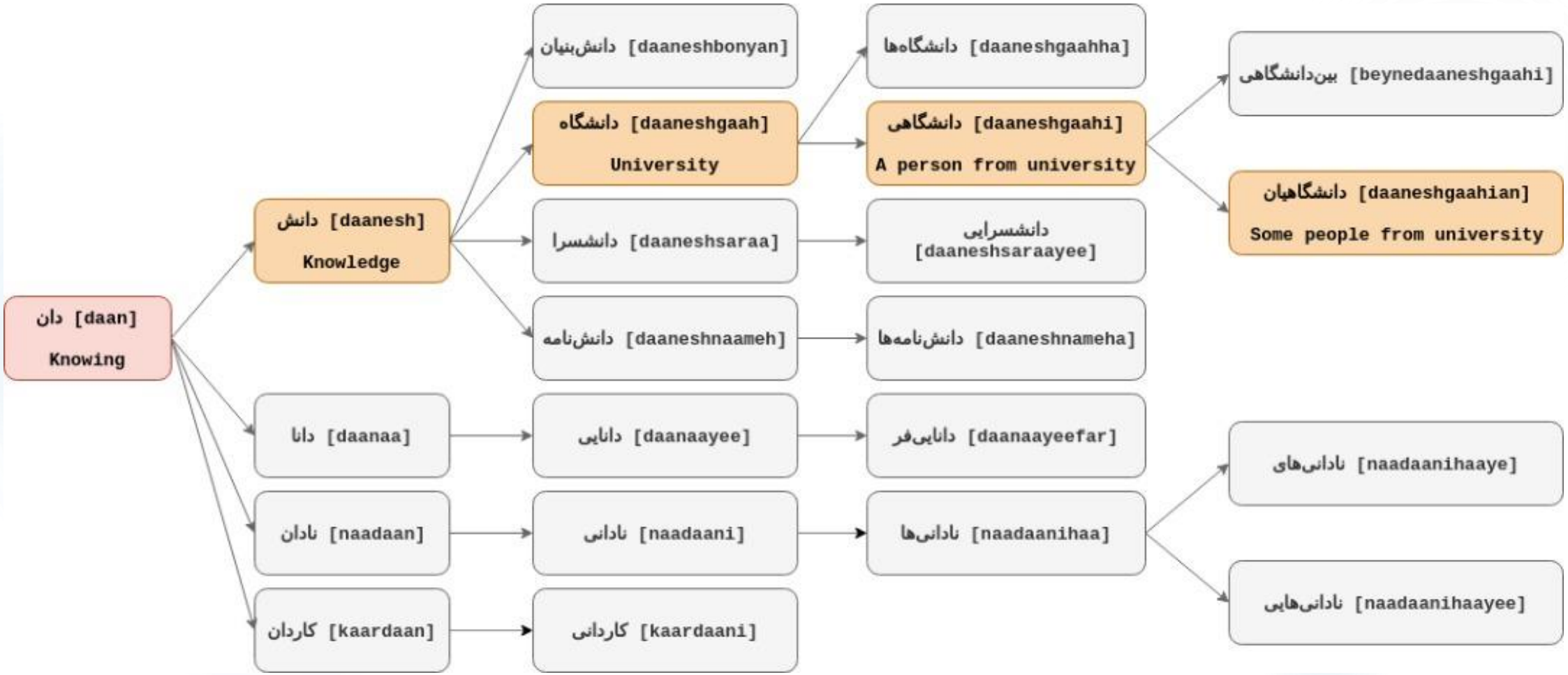
morphological network – definition

- one relatively novel type of morphological data resources are word-formation networks
- represents information about derivational/inflectional morphology
- in the shape of a rooted tree
- the derivational/inflectional relations are represented as directed edges between lexemes

morphological network (example) root دان [daan]: knowing



morphological network (example) root دان [daan]: knowing – cont.



selected language – Persian

- powerful and versatile in word formation
- having many affixes to form new words (a few hundred)
- an agglutinative language since it also frequently uses derivational agglutination to form new words from nouns, adjectives, and verb stems
- Hesabi (1990) claimed that Persian can derive more than 226 million word forms

selected language – Persian – cont.

- research on Persian morphology is very limited
  - Rasooli (2013) claimed that performing morphological segmentation in the pre-processing phase of statistical machine translation could improve the quality of SMT.
  - Arabsorkhi (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

selected language – Persian – cont.

since no Persian segmentation lexicon was made publicly available, we decided to create a manually segmented lexicon for Persian that contains 45K words

automatic segmentation tools MORFESSOR

- software for automatic morphological segmentation
- two versions:
  - unsupervised and semi-supervised versions
- more recent research on morphological segmentation has been usually focused on unsupervised learning

- an alternative: LINGUISTICA

data preparation

- primary sources
  - sentences extracted from the Persian Wikipedia
  - BijanKhan monolingual corpus
  - big Persian Named Entity corpus
- all data is pre-processed and tokenized
  - using HAZM tokenization toolset
- lemmatization of the data
  - tool presented by Taghizadeh et al (2013)
  - rule-based toolset proposed for this work

data preparation
semi-space in Persian

- a feature of the Persian and Arabic languages
- all semi-spaces are tagged by our software

word کتاب‌ها is the combination of کتاب and ها and could be written in two forms: کتاب‌ها and کتابها

data preparation
manual annotation

- words with more than 10 occurrences (97K words)
- distributed among 16 annotators (2 annotators per word)
- annotators made decision for:
  - segmentation (was accelerated by predicting morpheme boundaries by our automatic segmenting tool)
  - lemma
  - plurality
  - ambiguity (whether a word had more than one meaning)
  - removing if the word is not a proper Persian word

data preparation
manual annotation – removal

- when both annotators decided to remove a word, the word were deleted from the lexicon
- third annotators make decision about removal in case of disagreement
- after first step we had 55K words

data preparation
manual annotation – cont.

- if any disagreement happened, third annotator corrected it
- in some cases, some discussion to make the final decision
- all words were checked by the final reviewers
- final dataset: 45K words
  - 37K training set
  - 4k development set
  - 4k test set

data preparation – main problem ambiguities in written text

- the same surface form can represent different morphemes

- short vowels are not marked in written text, which results in different possibilities of analysis.
  - the word مردم *[mrdm]* could be analyzed, among other possibilities, either as the noun *mardom* (people) or as the past tense of the verb *mordan* (to die): *mordam* (I died).

data preparation
a snapshot

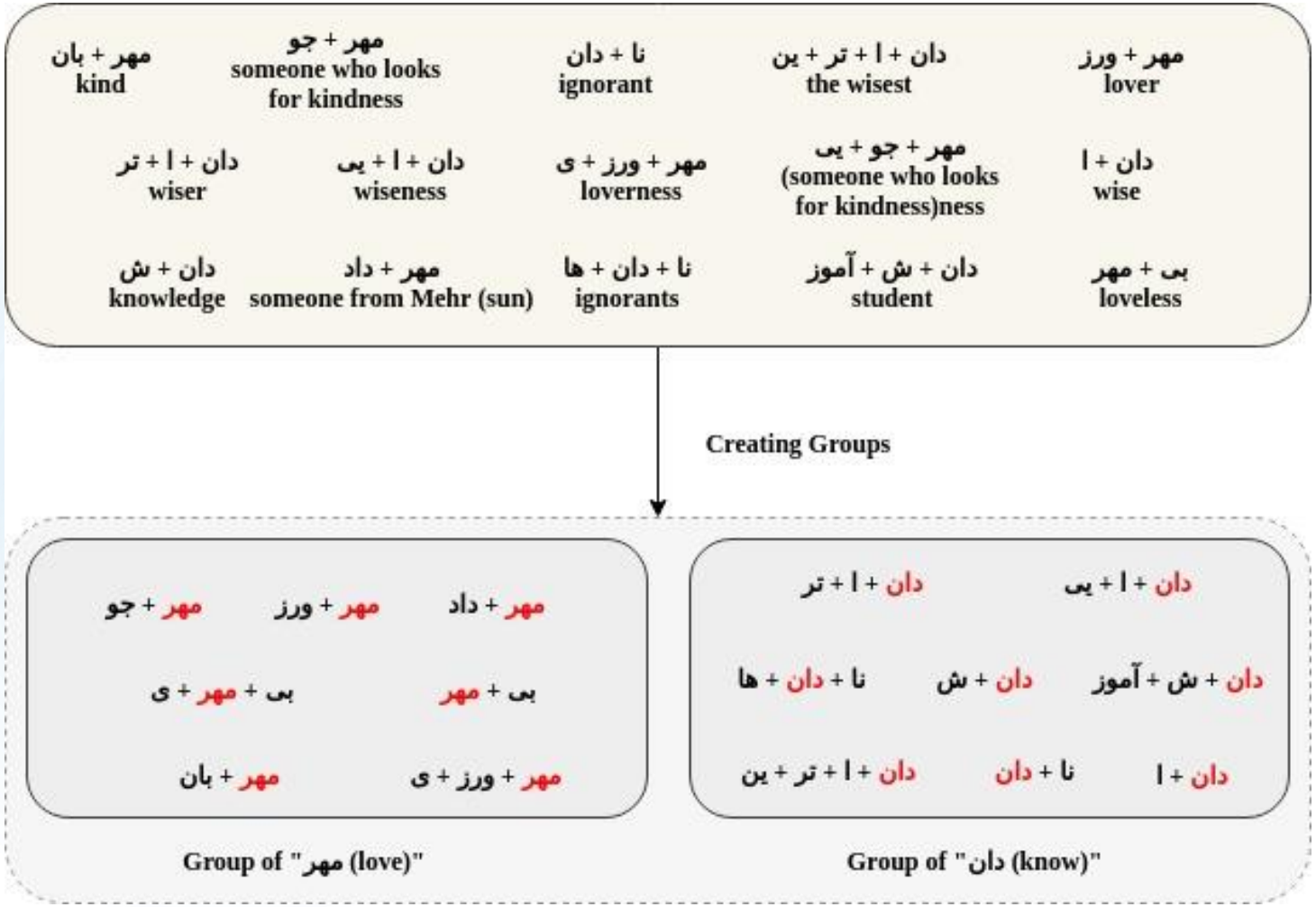
X	ملودی	**ملودی**	1		554		م		ل		و		د		ی						
X	ملودی	**ملودی‌ها**	1		43		م		ل		و		د		ی	X	ه		ا		
X	ملودی	**ملودی‌های**	1		147		م		ل		و		د		ی	X	ه		ا	X	ی
X	ملون	**ملونی**	1		20		م		ل		و		ن	X	ی						
X	ملوک	**ملوک**	1		439		م		ل		و		ک								
X	ملوکسیکام	**ملوکسیکام**	1		11		م		ل		و		ک		س		ی		ک		ا
X	ملک	**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		

morphological network construction automatic approach

main idea

- finding/tagging root morphemes
- grouping words based on predicted roots
- adding connections based on character overlaps

morphological network construction automatic approach – groups



two roots:

مهر [mehr]: kindness

دان [daan]: knowledge

morphological network construction automatic approach – overview

- phase 1: finding most frequent segments
  - 100/200: input parameter
- phase 2: removing segments (non-roots) from phase 1
- phase 3: group creation
- phase 4: tree construction for each group based on overlap length

morphological network construction automatic approach – pseudocode

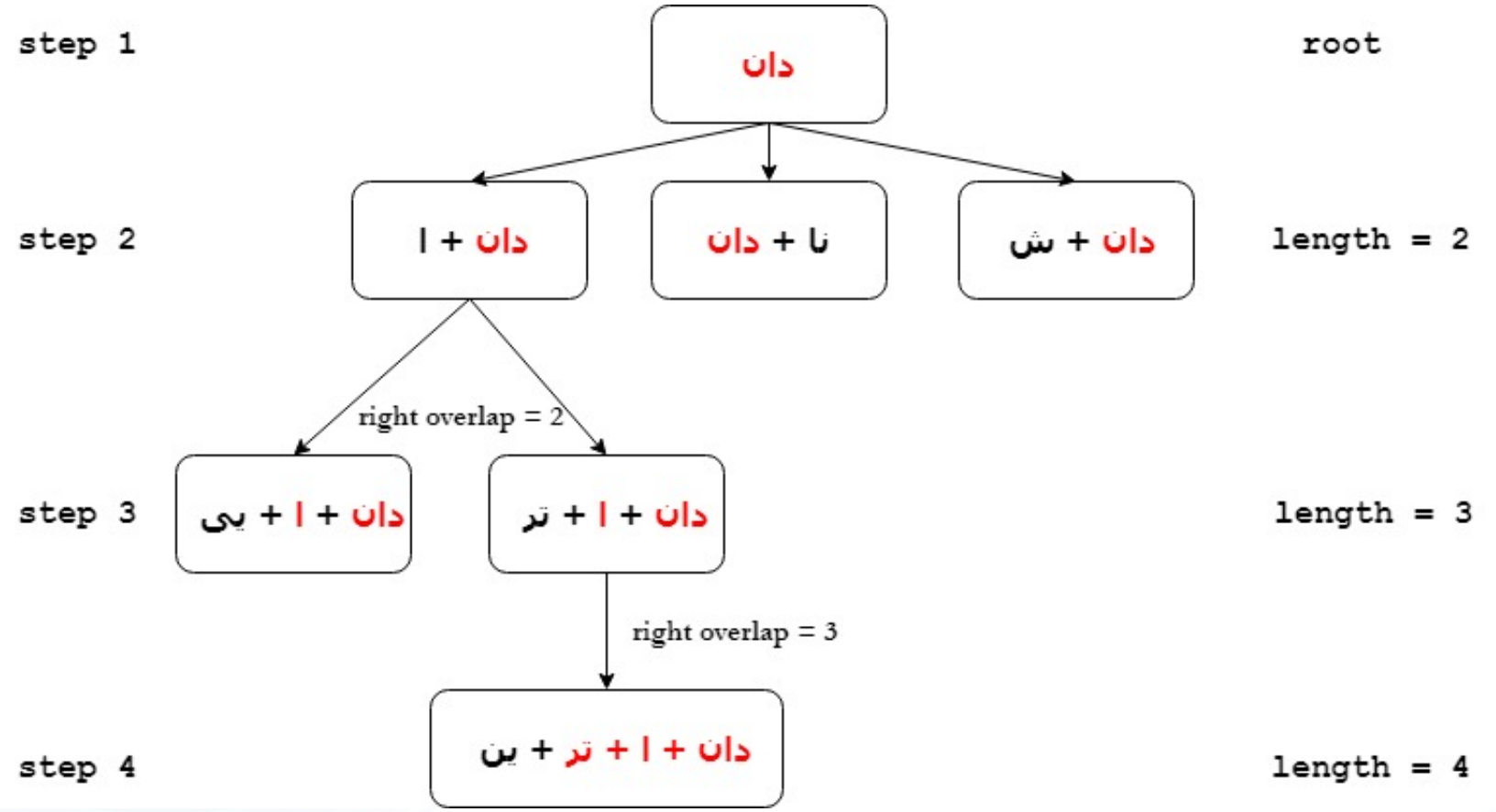

```
def generate(root, tree, words, n):
    tree[root] = root
    for word in words[n] and for leaf in leafs(tree[root]) :
        if overlap(leaf, word) > n:
            set_child_to_leaf(tree, leaf, word) and break
        else:
            remains.append(word)
        for leaf in leafs(tree):
   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

morphological network construction automatic approach – tree



automatic network construction example of non-roots

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

automatic network construction – example of non-roots – errors

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

morphological network construction automatic approach – recap.

- phase 1: finding most frequent segments (100-200)

- phase 2: removing segments (non-roots) from phase 1
- phase 3: group creation
- phase 4: tree construction for each group based on overlap length

morphological network construction semi-automatic approach – overview

- phase 1: finding most frequent segments (100-200)
- phase 1-2: checking most frequent segments manually
- phase 2: removing segments (non-roots) from phase 1
- phase 3: group creation
- phase 4: tree construction for each group based on overlap length

network construction examples from the real data

network construction results

results 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%

morphological network expansion goal – to increase the network

- from now, we want to increase the size of our network
- we can not increase the size of the segmented lexicon
  - it isn't an easy task
  - How much should we continue?

- using an automatic segmentation

morphological network expansion overview

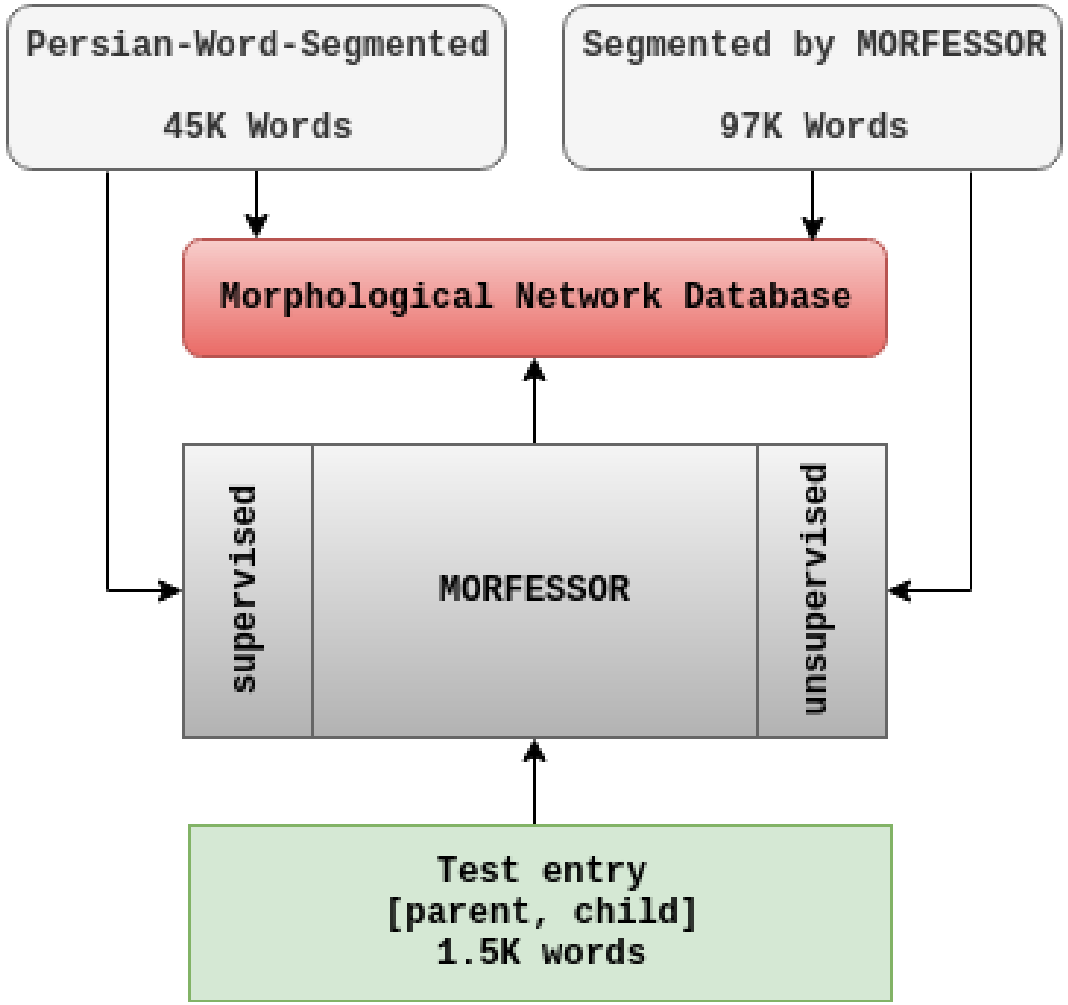
- phase 0: initial network is created (so far)
- phase 1: for new test word, the segmentation is done
  - using unsupervised MORFESSOR
  - using supervised MORFESSOR
- Phase 2: using the core algorithm the parent is found, the new word is added to the network.

1500 new test words are annotated for the evaluation.

morphological network expansion MORFESSOR

- unsupervised version: finding most frequent segments
  - 100K unsegmented lexicon
- semi-supervised version
  - 45K segmented words + 100K unsegmented lexicon

flowchart of our expansion methods



network expansion – results

accuracy for tree structures on 1.5K test 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

error analysis – network construction

- type 1: when a root morpheme considered as a non-root morpheme
  - discussed before
  - semi-automatic tree construction
- type 2: when a non-root morpheme considered as a root morpheme
  - morpheme "ون [oon] (not-common plural suffix)" was classified wrongly as a root morpheme

error analysis – network expansion

main error: wrong segmentation

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

data publishing

- in three different segments
  - training set: 37K
  - development set: 4K
  - test set: 4K
- the segmentation is done based on morphological network diversity
  - all word with similar roots are located in one segment
- data is available in LINDAT/CLARIN Repository:
  - **https://hdl.handle.net/11234/1-3011**

conclusion

- we created and introduced a new segmented lexicon for Persian
- we constructed Persian morphological tree
  - automatic tree construction
  - semi-automatic tree construction
- we proposed a tree expansion algorithm
  - unsupervised version
  - semi-supervised version

future plans

- using the unsupervised MORFESSOR to create derivational network
- using the supervised segmentation instead of MORFESSOR
- improving the data quality
- working on more languages: Turkish

დიდი მადლობა

Хвала

danke

cảm ơn bạn

hvala

ju faleminderit

शुक्रिया

gràcies

তোমাকে ধন্যবাদ

σας ευχαριστώ

آپ کا شکریہ

merci

با تشکر از شما

ధన్యవాదాలు

dankie jy

ありがとう

thank you

Дзякуй

gracias

நன்றி

takk

aliquam

谢谢

dziękuję

אַ דאַנק

شكراً

감사합니다

ขอบคุณ

eskerrik asko

grazie

děkuji

Terima kasih

спасибо

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

questions?